University of Cagliari

PhD School of Mathematics
and Scientific Computing

Perceptual Shape Analysis

Approaching geometric problems with elements of perception psychology

Author:
Fabio Guggeri

Supervisor:
Prof. Riccardo Scateni

March 15, 2012

Contents

Introduction

When teaching a machine how to behave like a human, it is mandatory to understand human behavior itself. It is impossible to describe an algorithm for the recognition of sadness if we don't know in first place what are the features that tell us a certain subject is, in fact, sad. The in-depth analysis of the psychology of human perceptions should then be the basis of every work that tries to give cognitive capabilities to a machine, and this is common practice in those fields directly linked to robotics.

There are, however, some fields where the cognitive process seems to be completely unrelated to human psychology, mainly because their area of interest developed as a sub-topic of a wider domain that originally had no connection to high-level tasks: this is the case of 3D Shape Analysis, aimed at the study of the high-level properties discernible from the surface of an object, that derived from the wider area of Geometry Processing. The strong focus on the geometric aspects in the latter field is reflected in the current state of the art of Shape Analysis algorithms, where the knowledge on a surface is pursued by means of complex mathematical calculations and every problem is approached in a low-level, numeric fashion.

This poses the motivation of this thesis: Shape Analysis is, nowadays, oblivious to the trend that can be seen in analogous research fields; *knowledge* and *meaning* are concepts familiar to the human mind and thus, in the author's opinion, should be obtained in a machine by mimicking the *algorithm* our minds perform naturally.

This work proposes possible solutions to a small number of standard Shape Analysis problems that try to reproduce the inborn human processes. The leitmotif of all proposals is that, despite the overabundance of information current algorithms can work on, human recognition is still better performing while lacking access to shape data like coordinates or curvature: the author's opinion is that the recognition process shouldn't then be dependent on such data, and while geometric solutions may offer interesting and useful insights on a shape, a high-level cognitive process cannot be based on a data-driven approach as it couldn't reproduce and imitate human behavior. Some topics, like *shape segmentation*, do take into account psychological and perceptual aspects in the problem, but just try to emulate the desired features in a data-oriented environment. It is then proposed a set of image-based algorithms that take inspiration from the visual system and the exhaustive work on perception psy-

chology, resulting in a more direct emulation of human vision and recognition. We refer to the proposed paradigm as **Perceptual Shape Analysis** (PSA).

The main claim is that, in addition to the high-level advantage of giving a human-inspired point of view, the new data-independent paradigm can obtain also low-level computational advantages as reduced time and memory consumption, robustness to noise and defective data and, last but not least, ease of implementation. It is shown that the results support our claims to a certain extent, and while some disadvantages cannot be ignored, the approach shows a strong potential and its complete novelty suggests that there is plenty of room for improvements.

The work is structured as follows: chapter 1 introduces the discussion on machine learning and human perception, with in-depth examinations on the perceptual processes that have a direct applicability to the field of Shape Analysis; chapters 2, 3 and 4 expose our proposals for perceptually based algorithms aimed at the resolution of three main problems in the field (respectively, the definition of shape descriptors, shape partitioning and shape reconstruction), along with a discussion on results, advantages and disadvantages for each implementation. Chapter 5 sums up the whole work, moving the topic from the up- and downsides of the single algorithm to the up- and downsides of the PSA, and suggestions for future extensions.

Chapter 1

Analysis of the human perception

1.1 The power of the human brain

The exponentially increasing computational power of hardware makes people wonder: when will computers outperform the human brain?

This question is still unanswered, and we can only make some predictions based on a comparison between the MIPS in a processor and the brain's neuron count. Hans Moravec [78] speculated that, according to the exponential curves of computational capabilities, computers would outperform human brains around the year 2020. Knowing however that the human brain is more than just computing power, Moravec moves the topic to a different level citing the famous case of the 1997 chess match between IBM Deep Blue and former world champion Garry Kasparov: for the first time a machine was able to defeat a reigning world champion under standard rules, making the event a notable milestone in the field of Artificial Intelligence. However, despite being just a collection of openings and endgames, Deep Blue gave Kasparov the impression of having the deepness of thought and creativity of a human player. This raises an interesting philosophical question: do machines really *think*? And what is, in substance, the act of thinking? Can we consider it just the mere application of algorithms hard-wired in our neural structures? Machines are nowadays outperforming humans in repetitive or numerical tasks, where little to none *thinking* is needed. Even the chess example is misleading: the game of chess involves a great amount of thinking for a human player, but the structure of the game and its limited, no matter how vast, set of possible outcomes make it easy to handle for a machine with enough knowledge on openings and endgames. But when it comes to learning, humans are still way ahead of their electronic counterparts: a computer needs a *descriptive* rule to solve a problem, while the knowledge of mankind is based on *explanatory* rules. Computers need to know what to do, we need to know the reason why.

1.2 Survival instinct and hardwired processes

There are however tasks that seem to be *hardwired* in our brains, the same tasks where no machine is at the moment capable of comparable results. Humans interact with the surrounding environment in such a natural way, thanks to personal experience, survival instinct and millennia of species evolution, that is difficult to code these behaviors into algorithms. First of all, the human is a social animal, and the social interactions are way too complex to be easily understood by a machine; moreover, most of the skills where the man is still performing better are directly related to the need of establishing social interactions: being capable of communicating, expressing and detecting small nuances in the speech, or recognizing facial expression are crucial skills when it comes to integrating in a society. In the year 2000 Ben Schneiderman [99] discussed how speech recognition was limited in its efficiency to applications with restricted vocabularies and constrained semantics, like command entry; the absence of *prosody* reduces the problem to a recognition of phonemes, with no emotional subtext or hidden meanings. In the last years many studies have been carried out in the recognition of those cues that make human-human interaction so complex: detecting emotions in speech [114] [113] is still an open challenge, especially when it has to be done in real-time for human-computer interaction, even if the machine can take into account also visual information coming from the facial expression of the speaker [11] [71] [21]. The problem is in general harder than a feature extraction: natural languages evolve in a cultural context that is shown to influence how the choice of words [87] and the perception of emotion itself [72] [26]. Even a textual communication, freed from the emotional information provided by the voice, could have to deal with allusions or innuendos that are hidden between the words: sarcasm [23] [107] [109], hostility [102] and humor [106] [76] are all peculiarities of a text that transcend the mere wording. All the cited works show how difficult it is to reach the accuracy of human perception in such topics.

Visual processes The field of human vision is not different: as sight is one of the most important sensory systems, it is strictly correlated with survival and instinctive processes. Nowadays many high-level Image Processing problems are being targeted in a fashion similar to the speech recognition problems, that is, localized to a single narrow goal: just as humor detection may be aimed at understanding whether a sentence is funny or not, but still lacks an effort on understanding *why* it is, problems like image segmentation [112] [56] [31] [1] can more or less satisfactorily extract regions of interest from an image, but leave the problem of understanding the contents of such regions to the field of object recognition [15] [65]. What we have is then a complex set of interconnected narrow fields that aim at a specific human capability and try to reproduce man's performance and efficiency. Anyway, the paradigm for most fields is, not surprisingly, to *take inspiration from the living*: if we want to teach machines how to behave like us, we first need to analyze and

comprehend our own behavior.

This work focuses only on the study of the shape recognition problem, from the acquisition to the high level analysis. The main goal of this work is to expose how humans parse and interpret shapes in order to transfer the processes into the field of shape analysis.

In two studies, Warrington and Taylor [120] [119] analyzed the recognition capabilities of people with unilateral brain damage, suggesting that object recognition is a two-stage process: people with right-hemisphere lesions showed an impaired ability in recognizing the same object when represented from different points of view, while a left-hemisphere damage was associated to the capability of recognizing the same object in the various stimuli, but also a deficiency in attaching a meaning to the object. This implies that object recognition, that is, the high-level task of assigning a meaning, is a subsequent step to the low-level viewing process; it is reasonable to focus then on how the information coming from the eye is manipulated in the brain, and what kind of descriptors can be extracted from it, before working on the *meaning* and *knowledge* coming from these descriptors. This perceptual hierarchy is reflected in the works of Marr and colleagues, exposed in the next section.

1.3 Silhouettes: more than meets the eye

David C. Marr was one of the most influential personalities in the study of the visual system. In his 1976 work *"Early processing of visual information"* [67] he exposes an analysis of the steps the viewing process is decomposed into, from the acquisition of the light stimuli to the extraction of information, focusing mainly on the definition of how the *meaningful* part of an image is extracted and separated from the *background*; the question whether this process is a bottom-up sequence of independent modules, or a complex net of interaction between different level of abstraction, is left unanswered, but poses the basis for his subsequent works: in his opinion, the process of recognition is hardware-independent, meaning that a good grasp on the dynamics of the human process itself is a necessary and sufficient condition to correctly implement the vision pipeline in a machine.

Marr notes that, despite the complex series of brightness values of a real world scene, an artist's depiction may consist solely of contour lines, suggesting that the correspondence between the real object and the artistic representation should lie in the artist's ability to extract a *contour* of the object itself from the complex scene; it means that a simplified, drawing-like representation of an image can be the starting point of the analysis of a scene. Marr finds however that despite the proliferation of edge-detection algorithms (as of 1976, Marr cites [91] and [39] among the others) the technique is still unsatisfactory; even nowadays, despite more than 30 years of advances in edge detection (most notably, the Canny algorithm [12]), no approach is yet capable of yielding descriptors of a high enough quality to capture the underlying knowledge in the image.

Figure 1.1: Picasso's lithography *Mother and child, with dancer and flute player* - 1962. The subjects can be easily recognized by a person even if the drawing is extremely stylized

The artistic drawing metaphor is found again in [68], where the author focuses on how such drawings are considered by the human viewer; we can easily give a three-dimensional meaning to the shapes in Figure 1.1, and even if the same silhouette can be obtained by an infinite set of possible shapes, our mind just selects the *most plausible* one, without ambiguity. This work focuses obviously on shapes generated by real-world objects, being *impossible objects*(like the famous Penrose Triangle in Figure 1.5) a category of shapes that have no application in the field of human and machine vision; an interested reader could however see the works of Huffman [40] or Penrose and Penrose [88] for a study of the interpretation of impossible shapes. The study on *occluding contours* [116] led Marr to believe that a silhouette is interpreted, under some restrictions (see Figure 1.2) as a **generalized cone** [7], theorizing that the mind follows a set of *a priori* assumptions (see Figure 1.4) on the shape leading to an unambiguous three-dimensional counterpart.

Richards et al. [90] approach the silhouette problem in a different way, creating a *codon representation* for the contours that describes how the shape curvature changes along its border (Figure 1.3). A codon indicates the inflections or maxima of the curve between two consecutive minima and the entire shape is then converted into a set of codons; the authors restrict the study to closed, smooth shapes described by a maximum of four codons in order to enumerate a limited set of curves, showing however that the remarks in their study can be generalized.

What's interesting to notice is that both studies result in similar descriptions of human interpretation under a similar set of constraints: according to Richards et al. all shapes are considered to be in a **canonical view**, that is, a general one where the shape is stable under small rotations and perturbations, and no dents, bumps or any other undulation on the 3D surface is expected when there is no evidence of such in the 2D contour, similarly to Marr's *gen-*

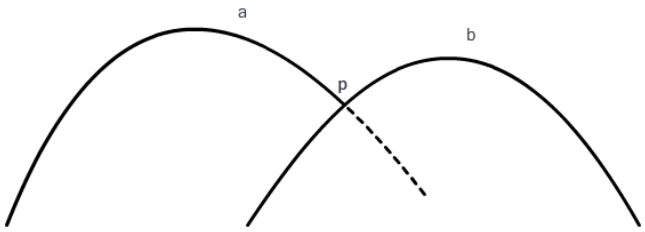


Figure 1.2: Marr poses some restrictions on what kind of silhouettes can be interpreted: 1) the curve must be smooth, there must be no overlaps, nearby points on the curve must be generated by nearby surface points, and the curve must be planar. The curve shown here violates the restrictions in **p**, and thus cannot be interpreted under Marr's theory

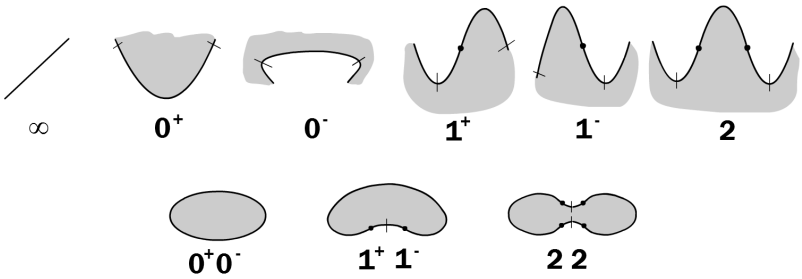


Figure 1.3: A codon describes how the shape changes between two consecutive minima of curvature (indicated by the slashes) according to the number of encountered inflections (indicated by the dots) and the sign of the curvature change. In the lowest row is shown the set of smooth closed curves identified by a pair of codons

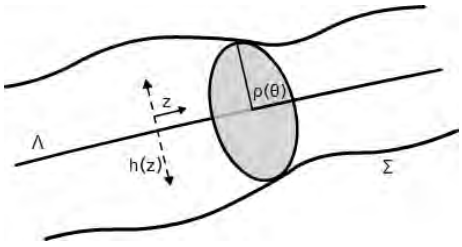


Figure 1.4: A **generalized cone** is obtained by sweeping a smooth cross-section $\rho(\theta)$ along the Λ axis scaled by a function $h(z)$

Figure 1.5: The Penrose triangle, one of the most famous impossible objects. The viewer interprets it as a 3D object, but cannot reconstruct a coherent shape

eralized cones where the shape changes smoothly according to its local radius and therefore all the surface perturbations should have a counterpart in the silhouette, and the constraint on the *vantage point* reflect Richards et al.'s definition of *canonical view*.

In conclusion, both studies pose a strong basis for a silhouette-based object recognition system; the absence of three-dimensional information is shown to be non influential, while other visual clues like shadowing can help improving the overall recognition but are optional. Moreover, the studies prove a set of strong correspondences between image contours and shape surface, most importantly the curvature sign, which has been used as the starting point of the consequent works on partitioning and analysis (see next section).

1.4 Parts and meanings

Giving an unambiguous and consistent 3D interpretation of a silhouette is the first step in the identification of an object; however, so far only *simple* objects and shapes have been considered, while the world is made of complex entities that our brain is able to acknowledge and identify. Marr and Nishihara [69] propose a set of criteria that descriptors used for shape recognition should comply to: **accessibility** specifies that a descriptor should be computable within a reasonable amount of time and accordingly to the limitations given by the images (resolution, quality); **scope and uniqueness** refers to the fact that a descriptor should be able to correctly and uniquely represent the whole class of shapes it's intended to describe; **stability and sensitivity** refers to the needed similarity of inter-species descriptors, along with the expressibility of small differences between the objects. With these criteria in mind, the authors study the usage of *stick figures* as shape representations (citing the works of Binford [7] and Blum [9]), proposing a hierarchical description using generalized cones [68] and their axes, as well as methods for the descriptor-based recognition; the cone axes are extracted from the image and the resulting description is matched in a database of shape categories in a top-down fashion, from the general class to the specific item. The construction of a *shape memory* is a crucial node in the whole approach, reflecting the role of the left cerebral hemisphere shown by Warrington and Taylor [119].

Hoffman and Richards [37], while maintaining the *shape memory* concept, approach the problem of complex objects by introducing the concept of **parts**: differently from previous works [81] [111] [32] [38] focused mainly on

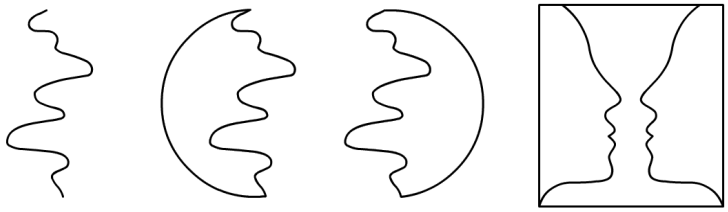


Figure 1.6: Reversing figures: the scribble on the left is interpreted differently whether the left part or the right part is considered as *foreground*. The famous Face-Goblet illusion (right) is based on the same perceptive principles

a primitive-based description of the object, the authors feel that a template-matching approach cannot perform well in presence of occlusion, where the proposed template has to be modified accordingly to produce a match. By partitioning the shape in different, *perceptually meaningful* parts, it is however possible to perform recognition only on the unoccluded parts. Another advantage should be that *parts* are best representative of non-rigid shapes like a hand, where a set of different templates must be used for matching all the possible gestures while a simple spatial relation between the different phalanges is consistent through every finger movement and helps limiting the size of the shape memory needed for recognition. Their definition of parts starts from considering the intersection between two objects and the surface curvature of such intersection, borrowing the **Transversality Regularity** principle from Guillemin and Pollack [35]: when two arbitrarily shaped surface intersect, they form a contour of discontinuity along their tangent planes. However the principle alone isn't enough to describe all the possible natural parts, that may be formed by *growth* rather than intersection (D'Arcy Thompson [108]); the authors observe though that the same regularity is visually applicable to all intersected or grown parts, extending the principle to shapes without discontinuities in the surface. They introduce the **Minima Rule**, stating that the boundary between two parts should lie on the minima of curvature in the surface: in the work is shown how the visual system intuitively interprets shapes according to this principle (see Figure 1.6). The Minima Rule is nowadays the basis for every shape partitioning algorithm (a discussion on these algorithms can be found in Section 3), each trying to separate parts of an object along the minima of the surface curvature and, despite being based on a rule describing how humans perceive, having little to no relationship with vision or perception approaches.

Singh et al. [101] extend the rule pointing out that the Minima Rule only states which boundary points should be used to partition, but gives no explanation on how to combine them into parts, especially when there is more than one possible configuration (see Figure 1.7). They introduce the **Short-cut Rule**, that states that humans tend to favor short boundaries instead of long ones, and therefore the boundary between two parts should consist of a line

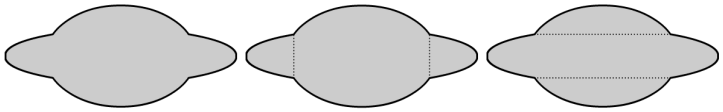


Figure 1.7: The Minima-Rule can cause ambiguous interpretations. Of the two possible interpretation, the Short-Cut rule states that the one on the right is intuitively better due to the shortest cut needed.

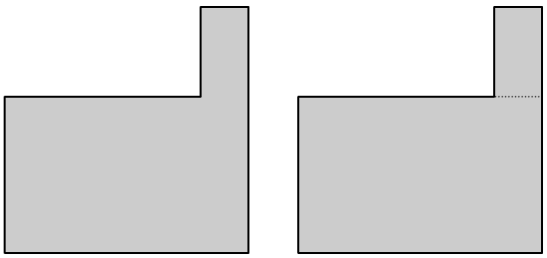


Figure 1.8: Another limitation of the Minima Rule: an intuitive cut doesn't necessarily connect two minima of curvature

segment (called *cut*) such that one of its endpoints is a minimum of curvature, it traverses an axis of local symmetry and is the shortest one among all cuts satisfying the previous criteria; they demonstrate that the rules reflects human perception thanks to a series of surveys, showing that the Minima Rule is not enough to describe how humans parse shapes as, for example, an elbow-like shape contains cuts where not every endpoint is a curvature minimum (e.g. Figure 1.8).

1.5 Multi-view recognition

A scrupulous reader could now object that the human vision system is stereoscopic, and the ability to perceive depth should be taken into account when constructing a recognition system that takes inspiration from the living. Anyway, as first showed by Marr, all the visual cues given by depth, color, shading and so on can be positioned on a higher level in the recognition pipeline; they do, however, add information to the process and can help in increasing the precision and efficiency of the process.

Orientation, for instance, plays a crucial node. Various studies [45], [73] showed that the time needed to name a figure is dependent the orientation; for *mono-oriented* objects, that is, objects that are usually seen and thought of by a single point of view (e.g. letters and numbers, monuments and so on), the recognition time increases linearly with the angular distance between the stimulus and the stored *canonical* representation. A study by Leek [55] focused on the recognition of the so called *poly-oriented* objects and their

long-term memory representation, where no canonical view can be chosen (Leek cites hammers, razors or pencils as examples). The study showed that the orientation effect observed in mono-oriented images stands also for poly-oriented objects, however, instead of being linear with respect to the distance from a single standard orientation, was influenced by a higher number of points of view, suggesting that a multi-view representation may be stored in the human memory. Going back to the *shape memory* concept described in the previous section, it is reasonable to think that a machine could work on an analogous set of multi-oriented shape descriptors to overcome the limitations of a single, two-dimensional projection.

Moreover, it is well known that a 3D shape can be reconstructed by its 2D projections: the entire field of Stereoscopic Reconstruction is based on this fact, and some of the principles are also used in the proposals in the next chapters; but aside from that, as the aforementioned field is outside of the scope of this thesis, many other works in the field of perception have been carried out towards reconstructing shape and orientation from silhouettes. Most notably, Murch and McGregor [80] proposed an algorithm capable of reconstructing such attributes from simple contours with no *a priori* knowledge on the depicted shape. This work, among all the others in the previously cited fields, show how a multi-view approach is enough to compensate for the lack of depth stereopsis.

1.6 Applied perceptual psychology

The literature in perceptual psychology covered every basic aspect of the recognition problem, from reconstruction to partitioning, showing how humans detect information on a 3D object just by analyzing its retinal projection. In all other machine learning environments, the design and definition of algorithms and programs is derived as a straightforward application of the psychological concepts, but this is not the case in Geometry Processing. Instead, as will be discussed in detail in the course of the next chapters, the trend seems to be deriving information from the *numbers*, with no interest on the processes that let our brains derive the same information but with a significantly smaller amount of data available.

This work aims at deriving a novel paradigm, a **perceptual approach** to the problems of Shape Analysis and Geometry Processing, to show how an accurate understanding of our behaviors is a vital part of reproducing our capabilities in the machines. All the proposed methods will reflect the observation of the works cited in this chapter, using no other information as those that would be available to a human eye. In the author's opinion, this completely new approach gives a fresh point of view on the research topics and, while providing useful and efficient solutions to some of the problems, can constitute an interesting starting point for future developments.

Chapter 2

Shape descriptors

> descriptor (**noun**): something (as a word or characteristic feature) that serves to describe or identify; *especially*: a word or phrase (as an index term) used to identify an item (as a subject or document) in an information retrieval system
>
> The Merriam-Webster Dictionary

The concept of *descriptor* is strictly related to Computer Science, as it can be noted from the dictionary definition. It is, however, something rather intuitive: the possibility to recall something by means of a concise description is useful and common also in everyday life. When speaking about someone in a set of people it comes natural to address him/her by some defining features (e.g. "the tall guy", "the blonde girl") instead of accurately enunciating every single possible description of the indicated person; it reduces the effort and the memory needed for recognition, especially when the set of reference is very wide. An appropriate example is the identification of mushrooms: as the cost for an erroneous recognition ranges from sickness to death, mushroom hunters rely on a set of standard characteristics for the correct identification. Fungal taxonomy is then based on, for example, cap shape, gill attachment, presence or absence of a ring in the stem, and so on: each of these features identifies (almost) uniquely a species of mushroom and hunters can (almost) safely assume the edibility of their harvest.

In the context of Shape Analysis, *shape descriptors* are of no difference: a compressed set of features that reduces the complexity needed to describe an object and provides useful and insightful information about it and its geometric properties. The range of descriptors used in literature is so wide it is almost impossible to list them all; it is however possible to summarize their common characteristics and uses. Mainly, descriptors are used for recognition and retrieval: it is reasonable to expect a descriptor to be able to discriminate between different objects and to return similar values for similar objects. However, it can also be desirable to express even the slightest distinctions between

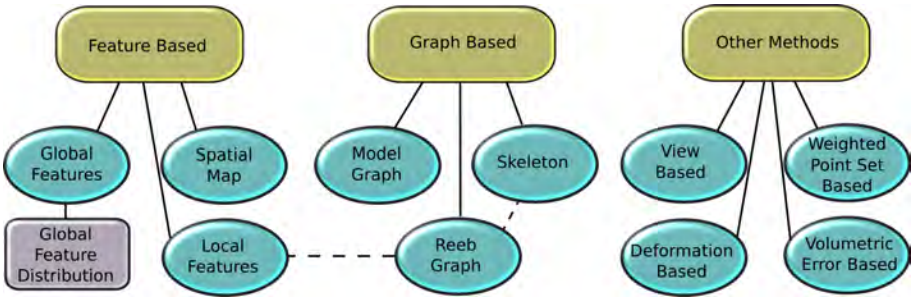


Figure 2.1: A taxonomy of shape matching methods according to Tangelder and Veltkamp [105]. The graph shows just one of the possible subdivisions, as many methods may fall in more than one category.

similar objects, and the tradeoff between simplification and precision is generally dependent on the application: if we need to extract birds from a set of animals, looking for feathers, wings and vertebrae can be enough, but it isn't sufficient for the detection of its species. This leads to another aspect of descriptors, that is, their efficiency is strictly related to the set of elements they're applied to: "the blonde girl" may not be a good way to identify somebody in Finland. Obviously, the more general a descriptor is, the better it should be: the efficiency of a good descriptor should remain mostly stable independently on the set of application.

There are then other desirable features that are relative to computational aspects, like robustness to numerical errors, discretization of the space or noisy shapes, roto-translational invariance, low computational complexity and so on. These characteristics will be analyzed in detail when presenting the algorithms that compute some particular descriptors, so the discussion on them will be skipped now. It is however interesting to show the common traits of the most relevant shape descriptors and the relative algorithms in the field of Geometry Processing, in order to introduce the discussion on how human perception can bring improvements to the subject.

Descriptors and geometry

A survey on shape retrieval by Tangelder and Veltkamp [105] approaches the discussion on Shape Descriptors by dividing them into three main categories. Because their focus is on *matching* techniques and not the mere description, some of the considerations in the paper may not directly apply to our topic, but as each matching algorithm is directly related to a descriptor or family of descriptors it is reasonable to follow their discussion for an introductory purpose. The authors present a taxonomy of matching methods identifying three main categories (namely **feature based**, **graph based** and **other methods**, see Figure 2.1) each approach could belong to; these categories are however not mutually exclusive, as some methods may have elements in common with

different categories, and the classification is then a graph rather than a tree. **Feature based** methods rely on the construction of a *feature vector* of fixed dimension d that describes the shape: the matching is then performed in different fashions, e.g. a *k-nearest neighbor* search in the d-dimensional Euclidean space, but this is a matter of matching and not description, and even if the two concept are related, it is beyond the scope of this discussion. What should instead interest us is the way the *feature vector* is constructed. Tangelder and Veltkamp subdivide the category in four main sub-fields according to what data this vector contains: **global features**, **global feature distributions**, **spatial maps** and **local features**. A **global feature vector** describes the whole shape by means of a combination of particular geometric measurements while **global feature distributions** focus on the statistical distribution of the vectors instead of comparing the feature values directly; **spatial maps** capture the spatial location of the object after a pose normalization, and **local feature vectors** aim to describe a particular primitive (typically a shape vertex) instead of the whole shape, and the object is then described by a set of vectors and not a single one like in the previous cases. The actual measurements are somewhat unrelated to the category and range above all kinds of geometric properties like volume [123] [124], statistical moments [123] [86], compactness [20], symmetry [49], angles and distances between surface points [84] [41] [85] or to the principal axes [83], spherical harmonics [50] [82] [115] and many many more. **Graph based** techniques extend the pure geometric description focusing on how the different shape components are related and linked; examples of this approach include *Reeb graph* based works ([75] [74] [126] as examples, more can be found on a survey by Biasotti et al. [5]), or *skeleton* based algorithms [103] [64] [42], whose construction will be discussed in greater detail later.

It has to be noted that all of the cited techniques heavily rely on geometric computations. In fact, except for a few works that Tangelder and Veltkamp list under the **other methods** (view-based [63] [22] or 2D sketch-based [29] retrieval methods among the others), there seems to be little consideration on how the human brain processes and identifies shapes. It has been established, in the works cited in Chapter 1, that no state-of-the-art algorithm is performing nearly as efficiently as a human in object recognition, and it is the authors' opinion that while all the cited features provide useful information that can easily be processed by a machine, they don't allow to directly implement the human behavior, which is, to this date, the best recognition scheme available.

So, the proposed paradigm follows a simple guideline: *try to imitate the human vision processes*. Rely mainly on data that would be available to the eye, avoiding to compute, unless strictly necessary, features that are dependent to geometry, primitives, connectivity and so on. We present on the next sections a revisiting of some state of the art descriptors in the perceptual paradigm. The discussion is limited to *curve-skeletons* (Section 2.1), but suggestions on the extension of the whole approach to different descriptors are present in Section 2.4, which will sum up the upsides and downsides of the paradigm applied to the construction of shape descriptors, while the discussions on the individual

Figure 2.2: A 2d medial axis or *topological skeleton* is the union of the centers of the maximal inscribed balls (left image). For arbitrary shapes, the skeleton may result very noisy (center image). *Pruning* removes the shortest, least significant branches to retrieve a robust shape description (right image).

algorithms is found at the end of the relative section.

2.1 Curve-skeletons

Among the various descriptors proposed in literature, *skeletons* have been found to be of great importance in many fields due to their versatility: the thinness and centricity features obtained by skeletons make them suitable to be used for motion planning, their ability to accurately summarize the topology of an object is used in shape retrieval, data compression and so on. The original 2D definition [8] states that a *topological skeleton* or *medial axis* of a shape is the locus of centers of the maximal inscribed disks (see Figure 2.2).

3D skeletons During the decades the term *skeleton* has been applied to every kind of thin, mono-dimensional, graph-like structure that best represents a given shape, especially in the 3D case where the direct extension of the definition (i.e.: the centers of the maximal inscribed balls), does not guarantee the mono-dimensionality criterion; even if Dey and Sun [24] provide a formal definition for *curve skeletons* in 3D, its exact computation is hard and unstable, thus making preferable an ad-hoc skeleton extraction method that satisfies the desired goal-specific criteria. A great proliferation of several methods of computation is then found in literature, where the resulting descriptors have features and uses different to one another; the next Subsection presents a set of the current state-of-the-art extraction methods, along with a discussion of the peculiarities of each.

2.1.1 Previous works

Previous work on skeleton extraction consists of a large number of methods and approaches due to their importance and usefulness in many fields. The heterogeneity of such fields makes it difficult to expose the previous works under a common point of view. It is however reasonable to subdivide the

methods in two main families according to the object representation used; in the volumetric category, the works are based on a discretization of the surface for extraction, while geometric methods perform the skeletonization directly on the primitives that define the surface. For an extensive survey on skeleton extraction methods, along with a discussion on the common characteristics of such techniques, one may refer to [19]. In the following we focus our attention only on some methods most relevant to our work and mainly published after the survey was written. An interested reader could find a deeper analysis on shape analysis and medial structures by referring to the books [28] and [100].

Volumetric methods

Most voxel-based methods take advantage of the discretized space and known topological constraints. Thinning-based methods tend to iteratively shrink the shape while maintaining the topological coherency in different fashions; [125] performs a hierarchical decomposition of the volume, thinning each simple sub-volume to extract the segments that form the final skeleton. In [66] and [118] the goal is to parallelize the thinning process while trying to maintain the topology. Methods based on the discretization of a function detect critical points in such functions in order to extract the skeleton; in [18] a repulsive force function is computed from the border of the object to the interior, detecting ridges as skeletal points. In [30] thinning is guided by the distance transform, similarly to [36] where the distance from the border is used to propagate a front with different speeds. The 3D Distance Transform is also used in [2] to extract the *centers of maximal balls* to reconstruct a thin skeleton. In [60] the thinning process is guided by a measure called *medial persistence* to increase the robustness. Most of these methods tend to be computationally expensive, sometimes fail to preserve the topology of the object and are strongly dependent on the model resolution.

Mesh-based methods

Mesh-based algorithms are a highly heterogeneous family: as skeletons may be used for shape analysis, shape retrieval, animation or matching, the methods for extraction vary strongly according to the goal. In [24] Dey and Sun provide a formal definition of curve-skeletons as a subset of the medial axis, introducing a function called Medial Geodesic Function (MGF) based on the geodesic distances between the contact points of the maximal balls. The skeleton is extracted as the singularities of the MGF. In [3] a Laplacian contraction is applied to the object with topological constraint for skeleton extraction and segmentation, in a manner similar to [13] where the Laplacian contraction can be also applied to point clouds in order to extract the skeleton. In [98] a deformable model is grown into the object to detect the branches from both meshes and point clouds, while in [59] the mesh is iteratively decomposed into hierarchical segments, computing a centerline compression error until a threshold is reached. The common drawback of these algorithms is that

they are all dependent on the number of triangles of the mesh, both in terms of running time and output quality; this obliges to fix a trade-off between efficiency and quality.

2.2 Perceptual skeleton computation

It is worth noticing how all the current extraction methods are, as previously said, heavily based on the primitives describing the object. The geometric approach is, in the author's opinion, the main common drawback of all current Shape Analysis algorithms, and the *leitmotiv* of the whole thesis is to imitate the human behavior, avoiding as much as possible to rely on data that the human observer would not have available, e.g. the vertex coordinates or the shape connectivity.

It may seem a paradox at first glance: computing the centerline of a shape without accessing the actual coordinates that describe it is a challenge that cannot be approached from a mere mathematical point of view. But let's put aside for a moment the geometric definition of the descriptors we're trying to compute, and let's focus on the actual use and meaning. A *skeleton* is a thin and simplified representation of a shape; as every child that had access to a pencil could show, a *stick figure* is a perfect and unambiguous representation of the human body. What's so special in a stick figure, that can be *computed* so easily by a child but still needs heavy definitions and computations for a machine? The answer can be found in the work of David Marr and colleagues. As already introduced in Chapter 1, there is a lot of psychological background regarding what humans perceive and what is *stored* in memory for future recognition. We will now analyze in more detail the connection between the perceived shape and the perceived skeleton, so that we could extend this connection to the computational field.

Recall how Marr [68] suggested that 2D contours are interpreted as projections of *generalized cones*. In absence of overlap in the projection, the axis of symmetry of a projected shape is the **actual projection** of the axis of symmetry of the three-dimensional shape (Theorem 5). Even if the absence of overlap is a strong constraint, and his work states that this property is valid for most of the real-world cases (see Figure 2.3), the ambiguity given by superimpositions on the image plane can be solved by looking at the object from many different points of view: it has been shown to improve the recognition capabilities in human observers [55], suggesting that multiple rotated projections of the object should be sufficient in detecting the features of the 3D counterparts.

We take advantage of these observations to derive a novel method for curve-skeleton computation, based on the re-projection of each 2D medial axis in space: according to Marr's theory, those portions of space whose projections correspond to medial axes, are candidates of being part of the cone's axis. The details can be found in the next subsections: our **main claim** is that this method overcomes the drawbacks of traditional algorithms, both voxel based and mesh based, as it goes beyond the limitations coming from a particular

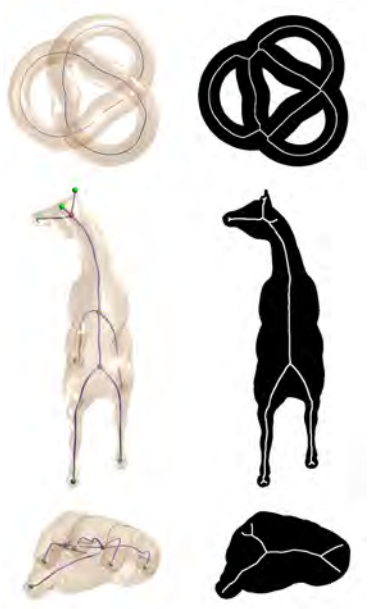


Figure 2.3: Most medial axes reflect the actual 3D shape when the overlap is reduced (upper, Knot model). According to the direction of view, some features may be missing (middle, Horse model), or completely unrelated (lower, Hand model).

shape representation. The perceptual background is robust to any shape that can be projected onto a screen, independently from its resolution; the proposed method obtains good results with coarse meshes and can quickly extract the curve-skeleton from fine meshes.

2.2.1 The algorithm

The proposed algorithm is, on a high-level, extremely simple and intuitive. We first implicitly compute an approximation of the Visual Hull ($\mathcal{VH}$)[1] of the object using a set of stereoscopic projections from different points of view. A medial axis is extracted for each silhouette and the spatial position given by the stereoscopic match is used to vote the corresponding voxel in a regular grid. Spurious votes are filtered out in the grid if they fall outside the $\mathcal{VH}$, and a maximized spanning tree of the grid is computed. The tree is pruned and processed according to an estimation of the radii of the inscribed balls obtained by the 2D information so that no artifacts remain in the final skeleton and the $\mathcal{VH}$ topology is preserved. Finally the skeleton is smoothed to improve its visual appearance.

[1] A detailed discussion on the Visual Hull can be found in Appendix A

Figure 2.4: To evenly cover the space around the shape we place cameras in the vertexes of a discrete 21-points hemisphere. The shape is centered in the origin of the frame, and each camera looks toward the origin.

The next subsections describe in detail each step of the algorithm.

2.2.2 Camera positioning and medial axis extraction

The choice of the viewpoints is the core factor in the construction of the approximated model of the object. Even though it could be possible to specify a mesh-dependent set of views [89], there is no way to understand whether the obtained $\mathcal{VH}$ is a satisfactory approximation of the shape [52].

We thus choose to evenly cover the space around the object, employing a regular grid of cameras centered in the vertexes of a discrete 21-point hemisphere. Covering just half of the visible horizon is enough since the silhouettes projection is symmetric. Both the shape and the hemisphere are centered in the coordinate reference system, while the cameras point toward the origin (see Figure 2.4). Intuitively, the more the viewpoints the more accurate the $\mathcal{VH}$. However, in our experiments, we found that finer resolution hemispheres do not increase the $\mathcal{VH}$ accuracy significantly.

To project points in the 3D space we then build up a set of 21 stereo acquisition systems, pairing each camera in the hemisphere with a second one, having direction of projection perpendicular to it. This direction is also parallel to the less principal component of its projection (which is given by the smallest eigenvector of its projection's Principal Component Analysis). This way of choosing the two directions of projection minimizes depth overlapping, and should, thus, be the best possible ones (an example of stereo pairs is in Figure 2.5).

For each projected binary silhouette we extract a Distance Transform (DT) based medial axis [93]. The medial axis is stable and reliable, and the DT

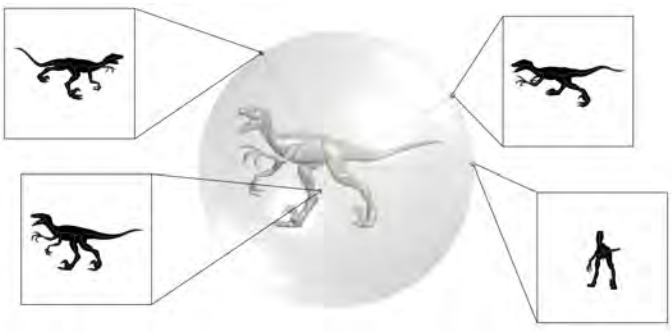


Figure 2.5: Two stereo pairs, color-coded, that include the point of view from which they are taken and the resulting silhouettes.

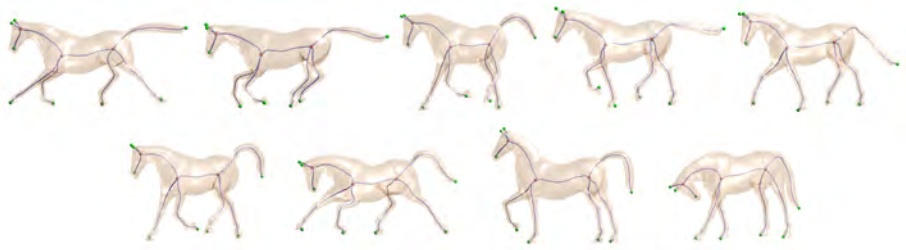


Figure 2.6: We show that extracting the skeletons of different meshes of the same object in different poses, we always obtain the same result.

values of each pixel give the thickness of the $\mathcal{VH}$ along the image plane, adding volumetric information to each stereo projection. The usefulness of this information will be evident in the following tree processing step.

2.2.3 Matching and radii estimation

According to Marr's theory a medial axis pixel corresponds to the projection of the desired curve-skeleton. This means that, when inverting the problem, each pixel is the starting point of a line in space, along the direction of view, where the 3D skeleton lies. If we re-project those lines in space, the positions with most intersections should correspond to the final skeleton.

Straight-forward intersections The simplest, straight-forward way to compute the intersections would be to actually cast rays from each medial axis pixel into a spatial grid, *voting* each encountered cell so that each voxel counts the number of images for which it is projected onto a medial axis pixel (see Figure 2.7). This approach however causes a series of issues that make it unsuitable for a fast and efficient computation. The main limitation is the computational

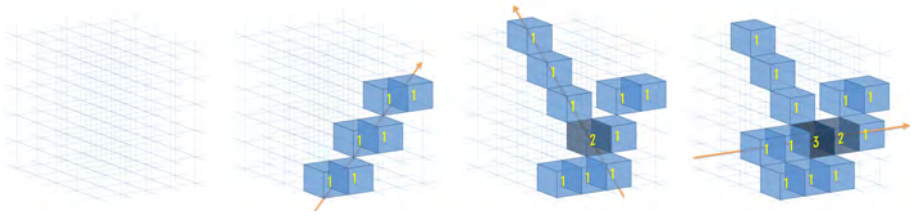


Figure 2.7: An example of ray casting, rasterization and voting

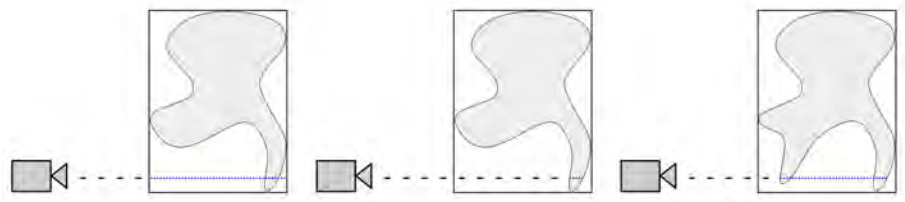


Figure 2.8: A complete rasterization is inefficient whenever the shape occupies a small portion of the bounding box (left). Depth-aware rasterization may reduce the waste of computation for some cases (center), but cannot guarantee an efficient procedure (right)

time needed for each ray, which is linear with respect to the grid dimension, as a *3D Bresenham line algorithm* is used for the grid traversal. Moreover, the portion of space actually occupied by the shape is potentially much smaller than the grid extent (see Figure 2.8, left), resulting in a waste of computational time. A *depth-aware* rasterization is possible if taking two projections along the same view direction from antipodal cameras, restricting the computation only on the portion of space comprised between the smallest and the greatest z-value in the depth images (Figure 2.8, center). The problem is however dependent on the shape configuration, and the superfluous computations cannot be always avoided (Figure 2.8, right).

Another possible approach is to reduce the rasterization to those pixels that have no overlaps in the projection by taking the GPU's *stencil buffer* and detecting the zones where the buffer has value 2 (that is, only a *front-face* and a *back-face* are rasterized in the pixel). However, this forces as a precondition the fact that the object has a watertight surface representation, strongly reducing the applicability of the whole approach.

In order to overcome all these limitations, a different approach is needed.

Stereo-matching intersections A more efficient way to detect the intersections would be to reduce the applicability field and, instead of computing many-to-many intersections by combining the information coming from all the images, focus on just two images at a time. In a fashion derived from the

field of stereoscopic vision and shape reconstruction, we combine two images in a system to obtain the line intersections in constant time. A way to simplify the matching is to employ parallel projections[1]. Two affine cameras P_z and P_x are positioned respectively along the z and the x axis, whereas the shape is centered in the coordinate system reference. Such cameras are defined by the homogeneous matrices

$$\mathsf{P}_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathsf{P}_x = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In such system epipolar planes are parallel and their normal direction is the y-axis. As the epipolar constraint coincides with the scanlines and projection rays are both orthogonal and axis aligned, back-projection becomes trivial: each pair of rays has the form

$$r \begin{cases} x = p \\ y = q \end{cases} \qquad r' \begin{cases} z = k \\ y = q \end{cases}$$

where $y = q$ is provided by the epipolar constraint and the complete separation between x and z coordinates is provided by the orthogonal directions of projection. The back-projected point is then

$$r \times r' = [p\ q\ k\ 1]^T$$

.

To keep the back-projection so simple we move the shapes instead of the cameras. Let's consider a shape S centered in the coordinate system $\mathcal{F}(O, X\ Y\ Z)$: given a set of stereo points of view v_1, v_2 we define a new coordinate system reference $\mathcal{F}'(O, X'\ Y'\ Z')$ where the Z' and X' axes correspond, respectively, to the lines joining v_1 and v_2 with the origin O. To get the projection we then bring S in $\mathcal{F}'$ applying the transformation $t^{-1}(S)$, where t is the rotation matrix defined such that $\mathcal{F}' \equiv t(\mathcal{F})$.

The method is based on a discretized voting space that can result in different vote accumulations depending on the coordinate system. In order to improve the robustness to rotational variance, we perform a Principal Component Analysis over the mesh vertices in pre-processing. We then rotate the object so that the first camera points toward the eigenvector corresponding to the smallest eigenvalue. This minimizes the information loss in projection. The *up-vector* is set parallel to the greatest eigenvector, thus maximizing data distribution on the y direction used in scanline matching. In this way we are able to define a pair of views where the xy image is supposedly the best representative of the shape, and tends to uniform the set of views used for similar objects even under rotation.

The stereographic view system can accurately reconstruct the original positions and radii of the medial axes balls when matching is one-to-one. When the matching is many-to-many, the direct xy to z matching results in multiple

[1]See Appendix A for an insight on *epipolar geometry* and *image rectification*

points and the level of confidence of both ball position and radius rapidly decreases as the number of points increases. We employ a multi-view voting system in order to give higher weights to those branches that remain consistent through a higher number of views.

The regularity of the grid makes the voting a simple operation. Let suppose we have a match, falling in the voxel i, j, k of the grid $\mathcal{G}$. The update procedure consists just in incrementing the correspondent cell

$$\mathcal{G}[i, j, k] = \mathcal{G}[i, j, k] + 1.$$

Each voxel stores also an estimated radius of the inscribed ball thanks to Distance Transform information on the generating images. Since the DT is an approximation of the triple of the Euclidean Distance, for each pixel we can compute an estimated distance of the skeleton to the border of the hull along the image plane. Let $\mathcal{R}[i, j, k]$ be the radius associated to the voxel i, j, k. For each new vote in that voxel, we update the radius as

$$\mathcal{R}[i, j, k] = \min\left(\mathcal{R}[i, j, k], \min\left(DT_{\text{front}}, DT_{\text{side}}\right)\right),$$

where DT_{front} and DT_{side} are the DT values in the generating pixels, taking the lowest radius estimate along all the views that contribute to that pixel. In this way we obtain a good approximation of the distance of each voxel from the border of the hull.

Each voxel in the grid $\mathcal{G}$ has a starting value of 0 and each estimate of the radius has a starting value of ∞. All the non voted voxels will not be taken into account by the further steps of the algorithm.

2.2.4 Grid processing

The low reliability of many-to-many matches may result in situations where spurious external branches stand out in the grid, especially in meshes with complex topologies or a high number of skeletal pixels in each scanline. Since we do not store an explicit $\mathcal{VH}$, but we define it implicitly using the silhouettes and the direction of projection, we perform a grid cleaning step where each cell is reprojected onto the images. If it falls outside a silhouette (i.e.: if the voxel is outside the $\mathcal{VH}$ of the object), it is set to zero. In this way, we get rid of the votes that are certainly spurious and do not have to be processed as skeletal candidates. This technique results in a strong improvement of the grid quality and, due to the low number of views and the simplicity of the operations, is computationally cheap and adds a little overhead to the whole procedure. In figure 2.9 we show a comparison between a cleaned grid on the Octopus versus its uncleaned counterpart.

Once we have the *voting grid* we can proceed with the extraction of the final curve-skeleton. The most voted voxels have the highest reliability in term of both position and radius, thanks to the higher directions of radius estimation and a higher centeredness in most views. We, then, choose to give higher weights to these voxels when computing the curve-skeleton. We extract

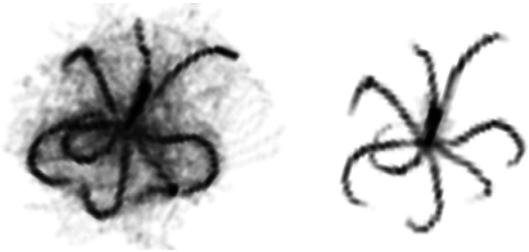


Figure 2.9: The cleaning process drastically improves the quality of the voxel grid.

a maximized spanning tree from the grid adopting a technique loosely based on the Ordered Region Growing (ORG) algorithm, described in [122], with several adaptations in order to fit the different kinds of structures we want to extract.

The ORG algorithm builds a tree-like representation of a 3D image (see Figure 2.10), where each voxel is a node and the edges between nodes form the path between two voxels. Such paths satisfy the least minimum intensity constraint, that is, the intensity in a path between two voxels is the maximum intensity achievable. Let $min(p_{ij})$ be the minimum intensity of each voxel in the path p between voxels i and j, and let g_{ij} be the path obtained by the graph traversal from node i to node j: it is guaranteed that $min(g_{ij}) \geqslant min(p_{ij})$ for every other p_{ij}. This feature is highly desirable in skeleton extraction, as the voxels with higher values in the grid are the best representative of the shape, due to a higher number of voting views, while low-valued ones should be used only for connecting different high-valued regions due to their expected inaccuracy or spuriousness.

The ORG tree is built as follows: starting from a seed point G_0 (the maximum valued voxel in the grid), the region G_1 is constructed from its 26-neighborhood, and edges between the seed point and each neighbor are added to the graph. Let G_i and B_i be, respectively, the region and its boundary at the i^{th} iteration, and s_i the maximum valued voxel in B_i. G_{i+1} and B_{i+1} are constructed from s_i by adding its unvisited neighbors, that is, those neighbors that are not already contained in G_i. New edges are created between s_i and each voxel in $(B_{i+1} \setminus G_i)$ and the process is iterated until every voxel has been included in the region.

2.2.5 Topological operations

After building the ORG spanning tree we process it to obtain the final skeleton. Let **N** be the set of all the nodes of the spanning tree, with **J** the subset of nodes with at least three incident arcs, **L** the subset of nodes with only one incident arc and **A** the set of all the arcs. Let $A_i \vdash N_j$ define that the arc A_i is incident on the node N_j. We define three topological operations that, applied to the spanning tree, give the final curve-skeleton. Such operations employ the definition of

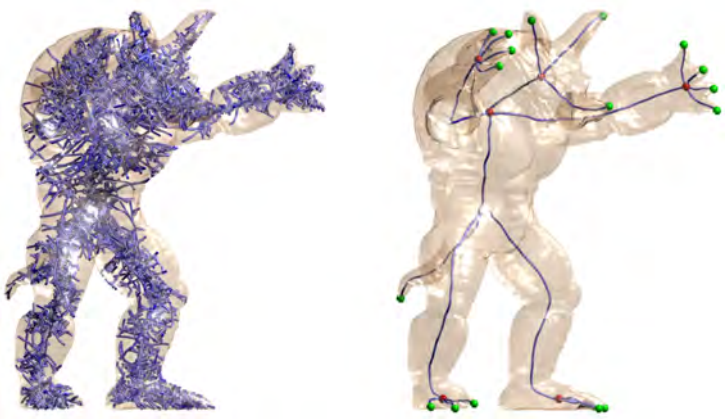


Figure 2.10: The ORG spanning tree (left) is an unorganized set of connected voxels. Only perceptually significant branches are extracted as part of the skeleton (right).

zone of influence (ZI) [95] of a node, that is, the volume defined by the maximal ball centered in it.

Perceptual core extraction

The skeleton consists of a very small subset of the spanning tree (see Figure 2.10), where the majority of the voxels have been voted as a result of spurious matches. The skeleton is extracted as the set of those nodes that are *perceptually relevant*, as human interpretation does not suppose the presence of dents or bumps in a shape without evidence (as shown in [90]). We, thus, discard the tree branches not projecting medial axes onto the images. In order to be perceptually significant, a branch endpoint must *stand out* in at least one view, that is, if and only if there is no intersection between its ZI and the ZI of the joint node its branch generates from:

$$\begin{aligned} &\forall A_k \in \mathbf{A} : \; A_k \vdash L_i \wedge A_k \vdash J_j \\ &\Rightarrow ZI(L_i) \cap ZI(J_j) = \emptyset, \qquad L_i \in \mathbf{L},\; J_j \in \mathbf{J} \end{aligned}$$

See an example of this operation in Figure 2.11a.

Branch collapsing

Often segments of the skeleton that are supposed to converge into a single junction point meet in different joints linked each other by short arcs. We, thus, apply branch collapsing, until convergence, for internal branches as long as there is intersection between the ZI's of each junction point:

$$\bigcap_{J_i \in \mathbf{J}} ZI(J_i) = \emptyset.$$

Merged joints will have as coordinates the barycenter of the junction points involved in the merging, and as radius, the minimum radius of the junction points involved in the merging. See an example of this operation in Figure 2.11b.

Loops recovery

If a shape has genus greater than zero, a subgraph of the spanning tree cannot represent its topology. In order to recover the proper topology we check the zone of influence of each leaf, closing a loop between all the endpoints whose zone of influence have non-empty intersection (see an example of this operation in Figure 2.11c):

$$\forall L_i \in \mathbf{L}, \forall N_i \in \mathbf{N} \setminus \{L_i\} \quad ZI(L_i) \cap ZI(N_i) = \emptyset.$$

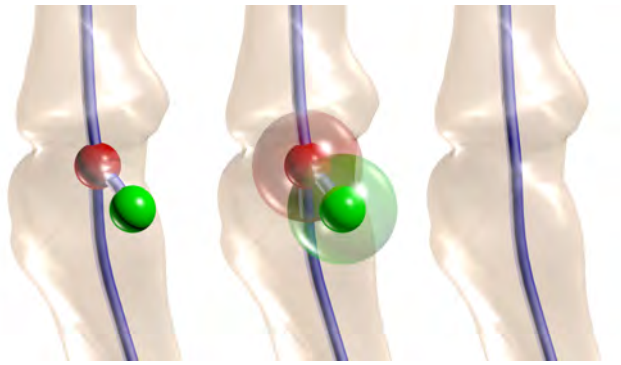
Note that the immediate neighbors of a leaf always satisfy the condition above, thus, to be sure that we really need to close a loop, an additional condition must be satisfied. Let p be the skeleton point whose zone of influence intersects the zone of influence of a leaf l. In the path joining p to l there must be at least one point having empty intersection with $ZI(l)$. This condition must hold for all the leaves involved in the loop closure.

To avoid the creation of fake loops, the topological operations described above must be applied in the order we presented them.

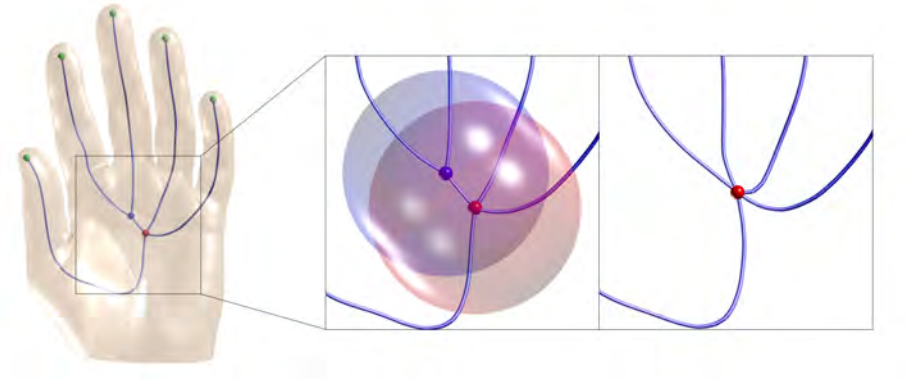
2.3 Results and comparisons

In this section we show the results of our approach in terms of skeleton quality, robustness and timings, along with some comparisons between our method and other skeleton extraction techniques. All the timings are obtained by single-thread implementations on an iMac with Intel Core 2 Duo, 2.66 GHz, 4GB RAM, and Ati Radeon 2600 Pro GPU.

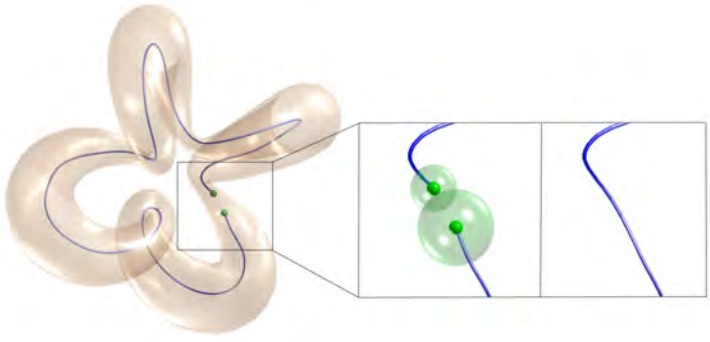
Our approach is capable of extracting correct skeletons that accurately reflect the topology of most kinds of meshes of any genus (see Olympics in Figure 2.12, Figure 2.3, and Figure 2.11c), even with complex morphologies like the Angiography and the Aneurysm models (see Figure 2.13), where the multiview system helps in solving the ambiguities caused by projection. The results are visually appealing and satisfy most of the expected criteria of curve-skeletons listed in [19]. **Homotopy** is achieved through the loop recovery operation described in Subsection 2.2.5. While there is no strict guarantee of correctness, the algorithm produces good results as long as the topology of the $\mathcal{VH}$ equals the topology of the shape and the estimated radii differ



(a) We prune a branch each time there is intersection between the ZI of its leaf and the ZI of the joint node its branch generates from.



(b) We collapse an internal branch each time there is intersection between the ZI's of its two endpoints.



(c) We close a loop each time there is intersection between the ZI of two leaves.

Figure 2.11: The three topological operations.

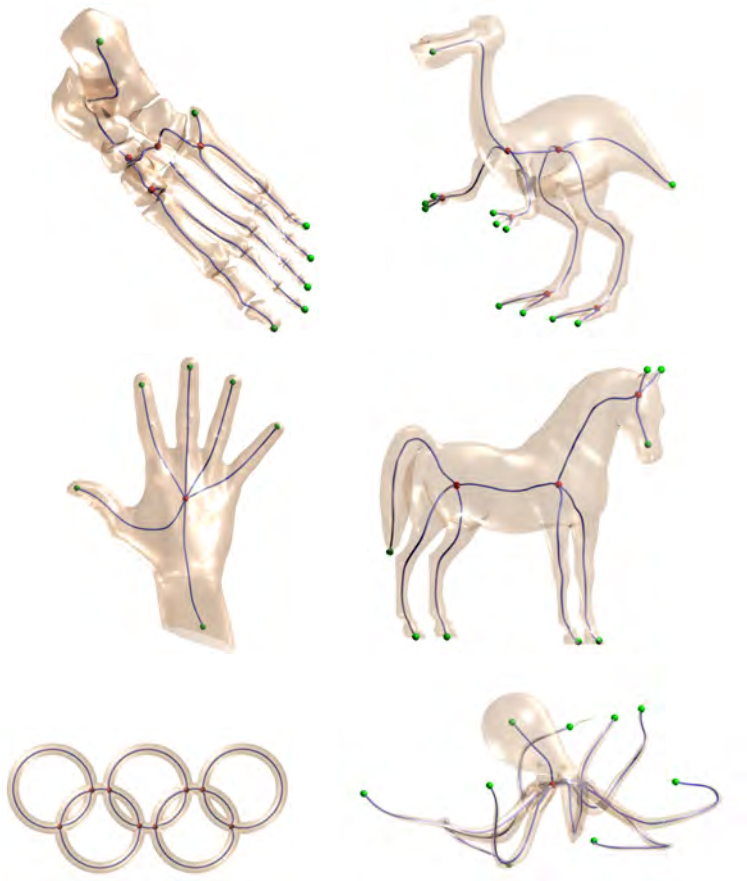


Figure 2.12: Curve-skeletons extracted (top left to bottom right) from the Foot, Dinopet, Hand, Horse, Olympics, and Octopus.

slightly from the actual inscribed balls. The tree extracted from the grid satisfies **thinness** and **connectedness**, while the algorithm is **robust**, **efficient** and guarantees **invariantness to isometric transformations**. Some properties are not fully satisfied. **Centeredness** is observed, but not strictly guaranteed. **Component-wise differentiation** and **hierarchy** are obtained in most of the objects, but the view-based approach has some limitations in capturing secondary junctions in some meshes, this behavior is discussed later. **Reconstruction** cannot be fully guaranteed by mono-dimensional descriptors like curve-skeletons, however, the union of the maximal balls centered in each skeleton point can produce a coarse shape approximation. Finally, our skeletons cannot satisfy the **reliability** criteria: no effort has been made towards direct shape-object visibility.

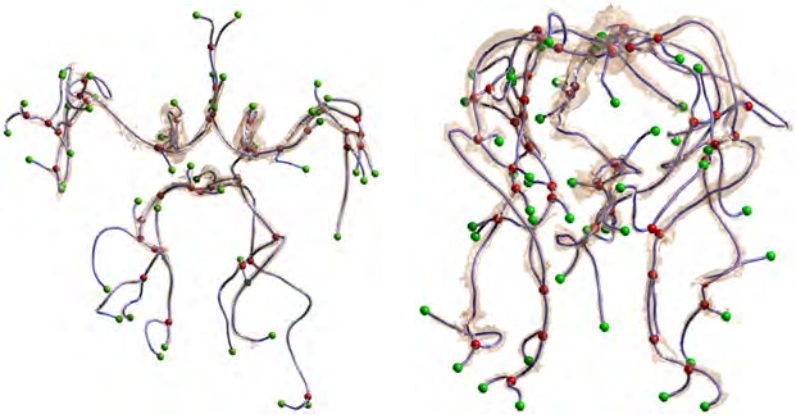


Figure 2.13: Two examples of curve-skeletons extracted from datasets with complex morphology: the ramifications of, respectively, normal and aneurysmatic vessels.

Our experiments show that the three resolution parameters (mesh, projection and grid) have little influence on the overall results both in terms of timing and output quality. While the overhead coming from bigger silhouettes is negligible and the grid resolution affects only slightly the ORG construction, the main computational bound is given by the shape projections: the mesh resolution influences the timings as more time is needed by the GPU to rasterize the primitives of the object. Quality wise, however, our method extracts coherent skeletons from simplified meshes, so it is possible to reduce the running times by decimating high-resolution meshes with nearly no information loss (see Table 2.1). The method is also unaffected by changes in grid and image resolution; Figure 2.14 shows how the different parameters affect the final computation; it can be noted that the skeletal structure is consistent and stable and, thus, is worth choosing low resolutions for both parameters plotted.

The projection approach makes the method very robust when used on

Faces	50%	25%	5%	2.5%	0.5%	0.25%	0.1%
Max error	2.19%	0.67%	1.53%	1.32%	1.48%	1.65%	1.87%
Avg error	0.16%	0.14%	0.17%	0.19%	0.31%	0.36%	0.57%

Table 2.1: Comparisons among skeletons extracted from the Raptor model (2M faces) at various resolutions. In the first row there are the decimation percentages. Each cell shows the Hausdorff and average error (in percentage of the length of the bounding box diagonal), showing that our output is mostly insensitive to strong decimation.

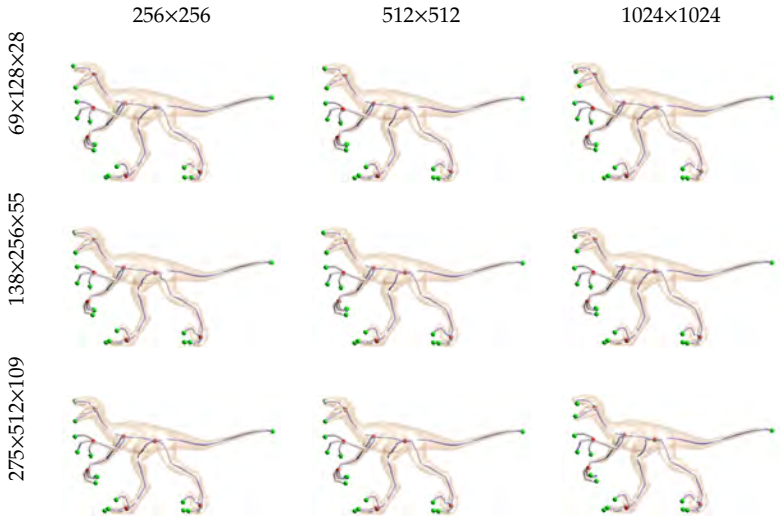


Figure 2.14: Results obtained at different silhouette (column-wise) and voxel grid resolutions (row-wise): the difference, in time, between the fastest (upper left, 0.31 secs) and the slowest (lower right, 5.02 secs) is one order of magnitude.

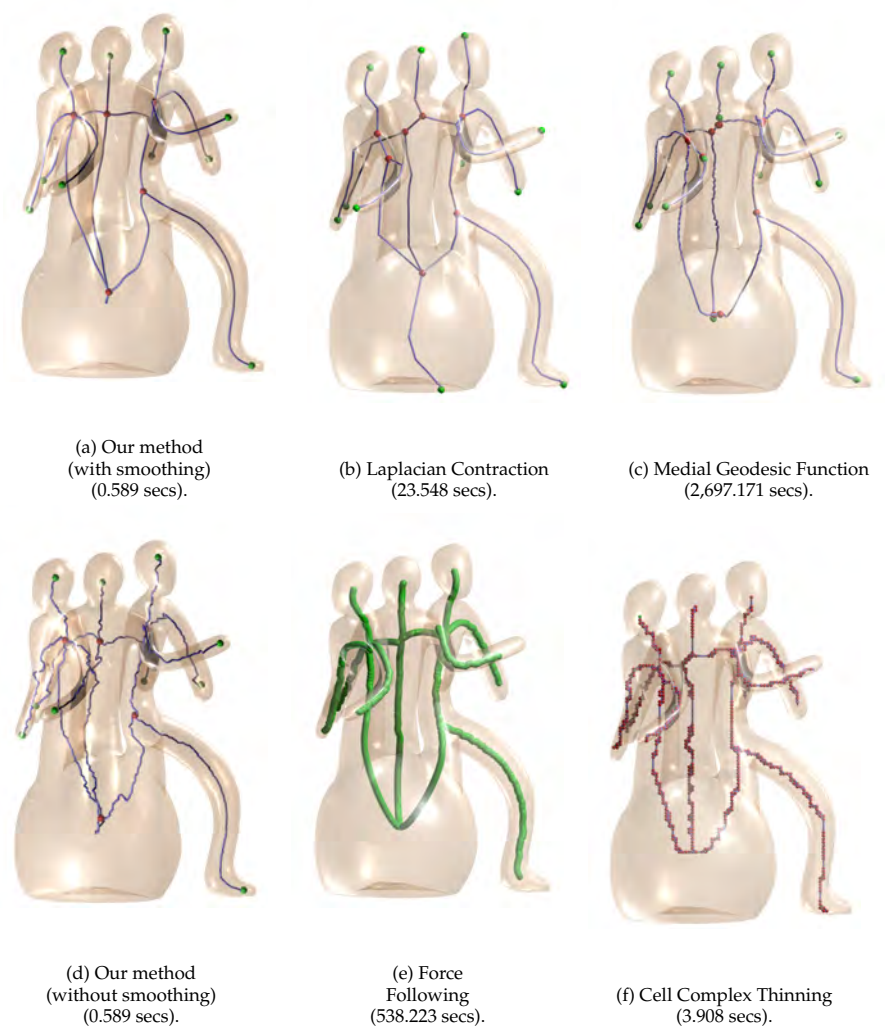
noisy data and even on incomplete ones. It is capable of extracting skeletons from non watertight meshes, as soon as their visual aspect is reasonable, since the holes are not influencing the production of the silhouettes. An example of these features can be found in Figure 2.16.

2.3.1 Extraction from raw point clouds

An appealing feature of our approach is its capability of extracting curve-skeletons from raw point sets (see some examples in Figure 2.18), as it needs no information about normals, thus differing from the majority of previous works in the field [104] which specifically need point clouds with normals. By performing a *morphological closing* of the projected image it is possible to reconstruct a silhouette that allows to proceed with the skeleton extraction. To obtain an accurate silhouette the size of the structural element must be chosen as a function of the density of the cloud, even if, for sparser point sets, narrow regions may be merged due to its higher size. However, the experimental results remain more than acceptable. The general benefits of the approach apply also to the point set case: the skeleton is noise insensitive and robust.

2.3.2 Comparisons

In this paragraph we compare our approach with four techniques cited in section 2.1.1; in the volumetric category we compare to the Force Following



(a) Our method (with smoothing) (0.589 secs).

(b) Laplacian Contraction (23.548 secs).

(c) Medial Geodesic Function (2,697.171 secs).

(d) Our method (without smoothing) (0.589 secs).

(e) Force Following (538.223 secs).

(f) Cell Complex Thinning (3.908 secs).

Figure 2.15: Visual comparison of different curve-skeletons extracted from the Memento model (52,550 faces).

algorithm [18] and Cell Complex Thinning [60], while we chose to compare to Laplacian Contraction [3] and the Medial Geodesic Function [24] in the mesh-based category. The timings, as listed in Table 2.2, show that our algorithm is noticeably faster than the state-of-the-art counterparts.

As for the volumetric methods, the main factor influencing timing is the thickness of each branch. The Hand mesh, for instance, even though has higher resolution than meshes like the Horse or the Octopus, is processed

Model (#Faces)	Our Method	Laplacian Contraction	Medial Geodesic Function	Force Following	Cell Complexes
Torus (1,536)	190	454	1,987	322,438	4,334
Octopus (15,054)	300	2,828	217,707	10,406	559
Dog (36,224)	341	10,675	554,937	51,500	2,123
Dino (47,955)	349	13,390	1,024,007	27,875	1,277
Hand2 (49,586)	450	18,172	1,434,891	86,109	2,282
Gargoyle (50,000)	522	19,328	897,839	180,938	4,677
Armadillo (69,184)	637	30,630	1,596,273	118,390	4,712
Horse (96,966)	585	41,765	3,294,194	49,516	2,024
Hand1 (273,060)	1,340	281,469	33,775,316	25,547	1,073

Table 2.2: Time comparison (in milliseconds) among the different methods tested.

faster, due to the thinness of its palm and fingers. A finer voxelization, needed for higher accuracy, then results in a strong increase in computational time, while our method is insensitive to the grid dimension. Parameters are a key factor also in terms of topological coherence. The Force Following algorithm, for example, may result in great gaps between the skeletal points, and a shape-dependent parameter tuning has to be found in order to obtain a coherent skeleton. The same can be said for the Cell Complex Thinning algorithm, which can produce both 1D and 2D skeletons, depending on the parameters setting. Our skeleton is unaffected by parameter changes, and failures in the topological reconstruction can be ascribed to a low quality estimation of the radii or to the $\mathcal{VH}$.

Comparisons with mesh-based methods show that our method is faster, especially for high-resolution meshes because of their dependence on the vertices. This dependence also affects the output quality when the resolution of the object is lower than a certain threshold. A coarse mesh with few vertices results in too few nodes for the skeleton or in an information loss (Figure 2.19, respectively center and right), while our method can accurately reconstruct the descriptor as long as its visual appearance is coherent.

2.3.3 Limitations

Our method is intended to work on character-like meshes as animals, human figures or cylindrical and articulated objects as tools, that is, the class of shapes where the Visual Hull is a good approximation of the actual shape. The multi-view system computes a skeleton of the $\mathcal{VH}$ of the shape rather than the object itself, making our algorithm unreliable for objects with no $\mathcal{VH}$ features (e.g.: the cup in Figure 2.20). However, a curve-skeleton may not be the best descriptor for such kind of objects in first place, where a surface skeleton like

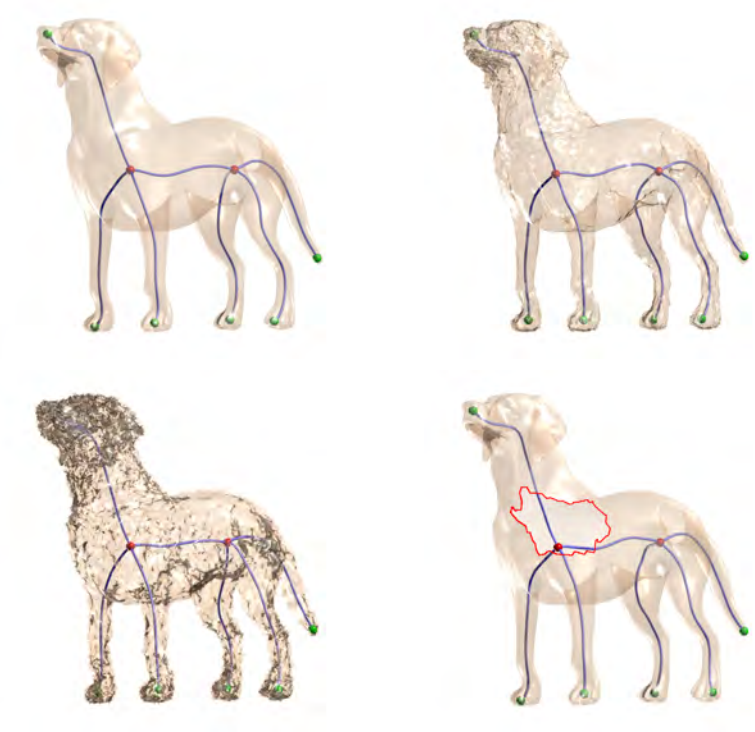


Figure 2.16: The projection approach leads to robustness under noise and incompleteness. We introduced increasing artificial noise in the upper right and lower left meshes, while in the lower right one we drilled a hole, highlighted in red.

[77] would better describe the shape when no protruding cylindrical regions are found. As long as it makes sense to choose a curve skeleton for the shape (e.g., for purposes of segmentation or animation), our algorithm is able to perform well.

Being based on the shape approximation given by the visual hull, the algorithm cannot extract all the features which are overlapped in **every** projection (e.g.: the buckyball molecule in Figure 2.20) or which are much smaller with respect to the projection resolution. In Figure 2.17 the ears of the dog remain visually close to its head in every silhouette, being ignored by our algorithm while detected in approaches like Laplacian mesh contraction. However higher resolution projections are able to isolate small features in at least one view, solving the problem. In our opinion, this reflects the behavior of human vision where the saliencies in an object are relative to the scale of observation [121]: a distant observer will notice less features in an object than a near one, while detecting anyway the most important parts (see Figure 2.14).

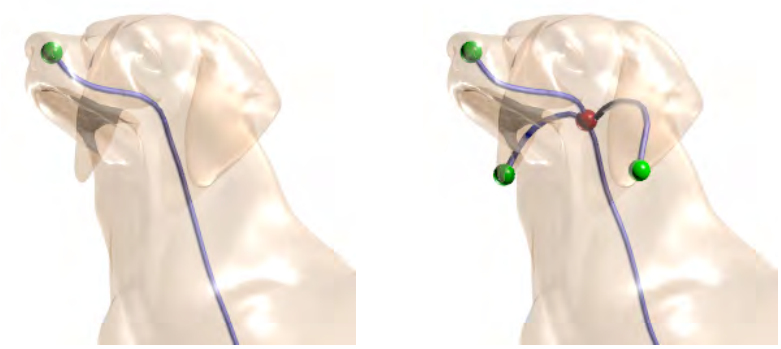


Figure 2.17: Two different skeletons for the dog's head, computed, respectively, with 256×256 (left) and 512×512 (right) pixel silhouette images. It is our opinion that the loss of fine details in low resolution projections reflects the behavior of human vision where the saliencies in an object are relative to the scale of observation [121].

2.4 Conclusions and future works

By taking advantage of the principles of human perception and stereoscopic vision, we have been able to propose in this chapter a novel approach to skeleton extraction, able to reconstruct a 3D curve-skeleton of the visual hull of the shape starting from the 2D medial axes of the projections of the object into the image plane. Reflecting the aim of the whole work, no object primitive is taken into account in the computation. The advantages of such an approach can be found in the independence on the mesh resolution or representation, while the results show also a good robustness to noise and missing data. The algorithm described in this chapter has been accepted for publication in [62].

The perceptual paradigm in other descriptors We would like to introduce a small discussion on the application of the new perceptual paradigm to other shape descriptors or measurements as a future extension. The whole ORG tree construction in Subsection 2.2.4 is necessary in order to extract a linear graph, however, when the desired descriptor doesn't require mono-dimensionality (e.g. a **surface skeleton**), the same set of stereo-matches can be used as a starting point. The estimated Distance Transform could be seen as an interesting approximation of the **local diameter** of the shape, in a manner similar to the Shape Diameter Function [97]. **Visual saliency**, computed by papers like [53], takes inspiration from perceptual elements while working on the mere geometry, while a perceptual-oriented approach could be more suitable. In the author's opinion there is a lot of room for perceptual algorithms in Shape Analysis. Every kind of descriptors that tries to emulate a visual process, or computes a feature that has an intuitive perceptual counterpart, could be ap-

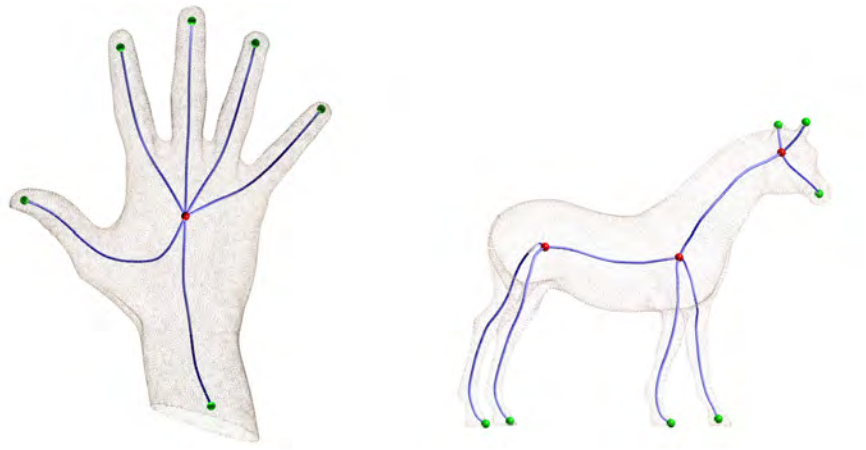


Figure 2.18: With our method we can extract skeletons even from unoriented point clouds. By performing a *morphological closing* of the cloud projections we reconstruct a set of filled silhouettes fed to the skeleton extraction for further stages.

proached in a manner that's similar to the one presented in this chapter. This is a refreshing point of view that could bring interesting developments in the future, along a path that has never been explored before.

(a) Our method. (b) Laplacian Contraction. (c) Medial Geodesic Function.

Figure 2.19: Skeletons extracted from a very coarse model (1000 faces). Primitive-based approaches cannot recover the underlying shape when too little information is available.

Figure 2.20: Two examples of datasets where our method fails: a model of a mug (top), and an isosurface of the buckyball molecule (bottom).

Chapter 3

Shape partitioning

> partition (**transitive verb**): **a:** to divide into parts or shares **b:** to divide (as a country) into two or more territorial units having separate political status
>
> The Merriam-Webster Dictionary

When dealing with complex structures, humans tend to subdivide them into smaller components, or *parts*, for an easier handling. It is a behavior that can be found anywhere, regardless of the application field: grocery stores organize their goods in categories making it easier to customers to find what they're looking for, or libraries store books accordingly. A long book is subdivided into chapters and paragraphs for an easier following, and even a *toy* task like copying a drawing is facilitated when the source image is inside a grid, so that one can focus on each single square instead of the whole picture.

The same *real world* advantages can be obtained also in Computer Science, where the definition of an independent, reduced area of interest can increase the speed of the computation, the memory usage and, last but not least, the easiness of coding; when applying partitioning in Computer Graphics, some works aim at a mere subdivision of the primitive into different disjointed sets for, e.g., parallel rendering [25].

However, the majority of Shape Processing algorithms aim at applying a common strategy on sets of primitives with common characteristics, so that the desired partitioning has to be *meaningful* in relation to the object: there is a need to subdivide the objects into *parts* that have a correspondence to the real-world components the object is composed of. **Segmentation** is the term mainly used for such a procedure, where each part is called segment.

But what is a *part*? How can we quantify the *meaningfulness* of a segmentation? As already pointed out in Section 1.4, the habit of partitioning seems to be innate in humans: the existence of a *shape memory* based on relational connections between subcomponents of an object has been theorized as an explanation for object recognition capabilities in humans and is supported by

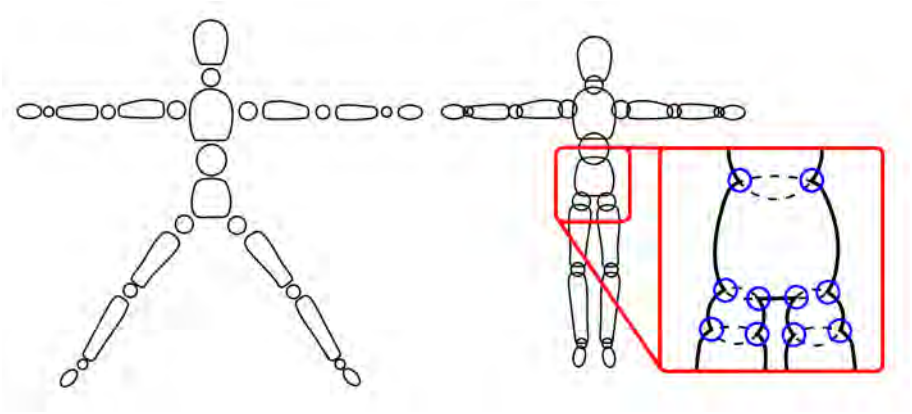


Figure 3.1: An intuitive explanation of the Minima Rule: a wooden lay figure is composed of several stand-alone convex parts (left) that, combined, give the complexity of a human figure (right). The compenetration of the parts give origin to concavities (circled in the closeup); a meaningful partitioning should then separate those minima of curvature.

the studies on early perception that show how the interpretation of shapes is dependent on a subdivision of its contour. Such studies, exposed in the next Section, became the starting point for every shape segmentation algorithm.

3.1 The minima rule and the short-cut rule in 3D shape analysis

The **Minima Rule** [37] and the **Short-cut rule** [101], already mentioned in Section 1.4, form a strong description of human interpretation. Their definition, even if intended for the contour segmentation, has been extended to the third dimension in order to give a *meaning* to the partitioning of a 3D object. According to the 3D minima rule, which is derived by the **Transversality Regularity** principle from Guillemin and Pollack [35], a discontinuity is found when two arbitrarily shaped surface intersect, resulting in a contour of negative curvature. The minima rule states then that boundaries between the parts should be placed in the minima of the curvature of the shape.

Intuitively, the definition makes sense: convex shapes are perceived as simple, stand-alone objects, while more complex object seem to be separable into different, more or less complex, sub-parts (see Figure 3.1). In the field of Shape Analysis, the minima rule has been taken into account in virtually every segmentation algorithm, even if combined with different metrics or methods of extraction. Here only a few are discussed, as the provided examples are enough to show the importance of the minima rule in the field of mesh segmentation. An interested reader may however refer to a survey by Shamir [96]

for further information on the subject. Katz and Tal [47] propose a hierarchical segmentation approach using *fuzzy clustering* between two faces to construct a fuzzy region that is then refined in order to minimize the curvature along the border. Similarly, Katz et al. [46] proceed by extracting the *core* of the mesh using a combination of its convex hull and a set of feature points on it, construct a series of patches and minimize the seams along the natural borders of the mesh. Liu and Zhang [61] use an affinity matrix defining the probability of two faces to belong to the same segment, apply spectral methods to cluster the faces favoring the segmentation along concave regions. As it can be noted, even if the methods approach the problem in completely different fashions, the minima rule is a common trait that cannot be overlooked.

Differently, the short-cut rule seems to have less relevance, as its contour-based definition makes it difficult to extend the concept to the three dimensions, differently from the minima rule where curvature has a surface counterpart. To the author's knowledge, only an attempt by Cheng et al. [17] tries to incorporate the short-cut rule in a skeleton-driven approach, minimizing the area of the cuts in a skeleton-driven framework.

3.2 Reconstructing the rules in 3D

Let's focus for a moment on human behavior in segmentation. We know from the cited psychological studies that a negative curvature in 2D is an indicator of a different *part* in the shape. We do, however, also know that the sign of the curvature in the contour is the same as the sign in the corresponding surface point (as long as Marr's restrictions are satisfied, Section 1.3). So we know where in the surface we should cut thanks to its contour, and the short-cut rule gives us an intuitive way of saying where such cut should end. Is it then possible to obtain a significative object segmentation by using only its 2D projections?

An automatic approach Experiments have been carried out in that sense. The idea was to find an automatic contour-based segmentation scheme and apply the multi-view approach to gather as much information as possible around the object to reconstruct the 3D cuts. In particular, two works for contour partitioning have been studied: one titled *Approximate Convex Decomposition for Polygons* by Lien et al. [58] and *Parts-Based 2D Shape Decomposition by Convex Hull* by Wan [117]. In the context of **Approximate Convex Decomposition** [57], a partition strategy defined by Jyh-Ming Lien for arbitrary shapes, the application to 2D shapes resulted in an interesting implementation of the perceptual principles. By defining a quasi-convex partitioning of a polygon, the approach gets rid of the most significant curvature minima around the border, while the flexible convexity constraint increases the robustness of the results. Wan's work, similarly, extends the quasi-convex decomposition using information coming from the shape's convex hull to reduce the number of spurious cuts. However, when aiming at reconstructing the partitioning into

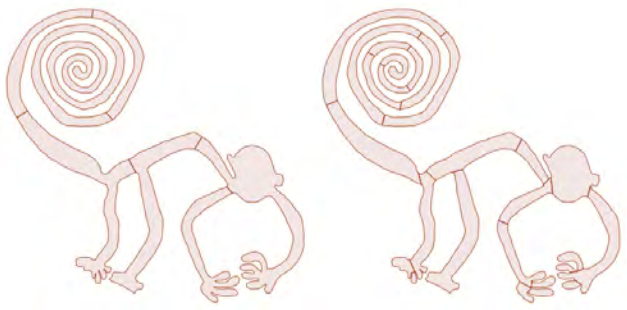


Figure 3.2: Automatic 2D shape decompositions (Lien et al. [58] on the left, Wan [117] on the right) don't reflect the partition a human would perform, and therefore aren't suitable for an extension to 3D shape segmentation

a 3D shape, both methods are unsatisfactory due to oversegmentation: the automatic approaches resulted too sensitive to perceptually unimportant curvature like in the tail of the Nazca Monkey (see Figure 3.2), where the inside of the spiral, of almost constant negative curvature, is the starting point for a number of unneeded cuts in both approaches, no matter how the parameters are chosen. Moreover, as expected, spurious cuts are found whenever the projection overlap forms angles in the contour. Differently from the skeletons, the cuts don't remain stable after small view rotations, and the overall method is unreliable.

3.2.1 A manual approach

A completely automated approach is, then, unaffordable at the moment. The stability of the cuts and their actual significance are too low to be reliable for an unsupervised segmentation. However, the principles described in the previous subsection can have an interesting application to the routine of *manual segmentation*. Manual segmentation is a technique for shape partitioning that tries to take into account the human factor in the most direct way possible: as the name itself suggests, a user is asked to manually indicate the parts that compose the shape. The typical usage of a manual segmenter is to construct a *ground truth* dataset for the evaluation of automatic segmenters, as in the Princeton Segmentation Benchmark [16], where the outputs from various algorithms are compared to a set of manually segmented shapes that should reflect the actual *significance* of the parts. Obviously, in order to construct such dataset, a user needs some kind of application that helps him/her define the parts on the shape without having to specify a segment index for each single face in the mesh. The main objective for a manual segmenter is then to limit the needed user input, letting a human *correctly* segment an object with the least amount of effort. Typically, the user specifies the position of a border by dragging the mouse over a region, and an automatic refinement step determines the best segmentation according to the surface features. The

Figure 3.3: Examples of interactive segmentation approaches (from top to bottom: Ji et al. [44], Fan et al. [27], Lee et al. [54])

way the user interacts may vary from approach to approach: in Ji et al. [44], for example, the user is asked to *scribble* the position of two neighboring segments (Figure 3.3, top row); Fan et al [27] prompt the user to *paint* on the mesh (Figure 3.3, center) the segment, while updating the borders to best fit the surface behavior; Lee et al. [54] propose a semi-automatic approach where the user may specify a cut on the image, whose border is refined automatically (Figure 3.3, bottom row).

Manual cut segmentation Our proposal is to let the user directly specify the cuts in the 2D projection, similarly to [54], trying to reproduce the perceptual principles that weren't suitable for the automatized approach. The user, after rotating the object into a desired view, drags a straight *cut* through the contour of the shape (see Figure 3.4). The drawn line is considered as a projection of a *restricted* cut plane in space; the software tries then to reconstruct the cut. In order to keep the procedure compatible to both meshes and point sets, no real face-plane intersection is computer, rather a set of *influenced* vertices, that is, those vertices whose distance to the plane is less then a certain threshold. Vertices on different sides of the cutting plane are given different segmentation indices and the indexing is *propagated* into the object to create a new segment (see relative paragraph). However a single cut may not be enough to define a patch boundary: the user is asked a second input to refine the cut.

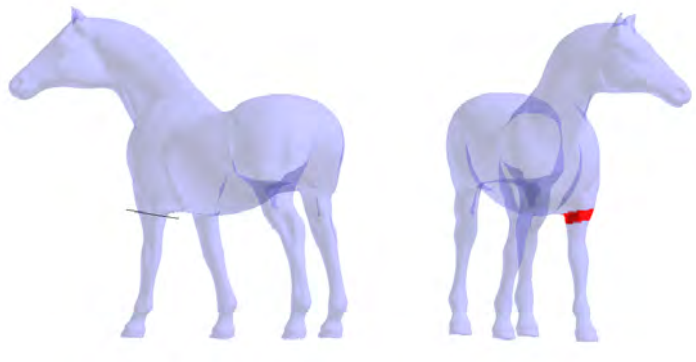


Figure 3.4: An example of cut definition: the user draws a line on the projected object (left) defining a plane that cuts the object (red vertices in the right image)

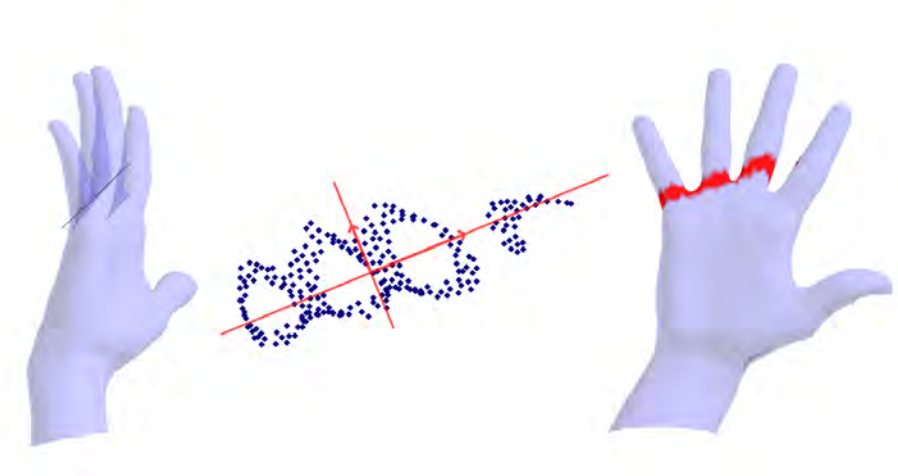


Figure 3.5: Automatic best-view selection: after a first cut (left), the vertices within a certain distance from the plane are projected onto the plane and the camera is rotated to match the smallest Principal Component

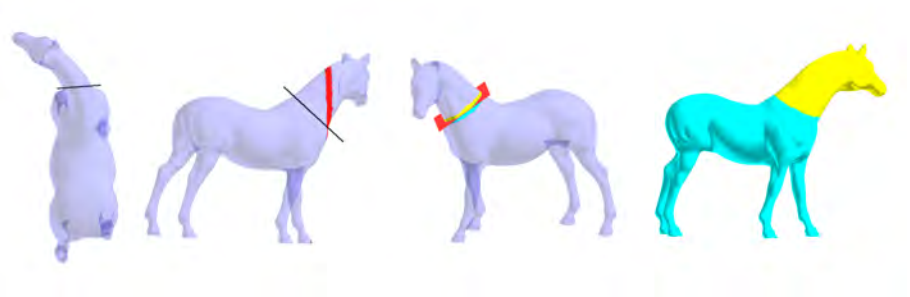


Figure 3.6: An example of two-step cut definition: an initial cut (first image) is combined with a second one (second image) to create the definitive plane (third image). The segmentation is propagated to the rest of the object (fourth image)

Best-view cut refining A single cut may be enough whenever the shape projection, according to the chosen point of view, has no overlaps in the cut region. A single connected component is extracted, the cut is confirmed for index propagation (see next paragraph). However, the single-view system is error-prone: it is possible that the cut returns a single connected component, but extremely elongated due to an incorrect orientation, or that more than one component is found (see Figure 3.5). Either ways, the cut must be further restricted to a region of interest. The best way to refine a cut is then to automatically select the *best* view that separates the intersecting components: every vertex influenced by the cut is projected onto the plane and a 2D Principal Component Analysis is computed on the projected points. The direction relative to the smallest eigenvalue is taken as the new view direction, thus maximizing the separation of the components and giving the user the best view to refine the cut. This also allows for a recovery for skewed cuts, as the secondary plane direction is used to correct potential errors in the orientation of the first plane. Suppose the user defined the first cut on an image I_1, obtained thanks to a projection along the view direction $\mathbf{v}_1$, with image coordinates $(x_1^{'} y_1^{'}), (x_1^{''} y_1^{''})$; while a plane p_1 can be obtained by back-projecting the cutting segment along $\mathbf{v}_1$, what the user is actually defining (due to the ambiguity given by the 2D projection) is that the normal of the cutting plane should be, when projected onto I_1, coincident with the normal direction of the cutting segment. This observation stands similarly for a second cut on a different direction $\mathbf{v}_2$. Let M_1 and M_2 be the transformation matrices from respectively the first image space and the second image space into object space, let $\mathbf{c}_1 = (x_1^{''} - x_1^{'}, y_1^{''} - y_1^{'}, 0)$ and $\mathbf{c}_2 = (x_2^{''} - x_2^{'}, y_2^{''} - y_2^{'}, 0)$ be respectively the direction of the first and the second cut transformed into object space, the cutting plane normal $\mathbf{n}$ is computed as

$$\mathbf{n} = (M_1 \mathbf{c}_1) \times (M_2 \mathbf{c}_2)$$

while the plane offset is computed such that the intersection of the two

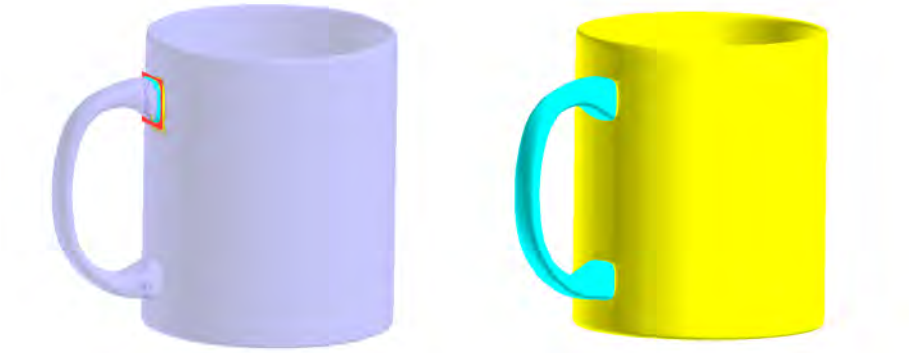


Figure 3.7: Whenever a single cut isn't enough to extract a segment, the application asks for a second cut on the handle

planes p_1 and p_2 lies on it. The final plane must be *restricted* to a quadrangle given by the image cuts. Each cut endpoint on the i-th image defines a delimiting plane having normal $M_i\mathbf{c}_i$; the intersections of such planes create four lines in space that intersect the cutting plane onto four points that are taken as the vertices of the final cut quadrangle that limits the segmentation border (Figure 3.6).

Index propagation After a cut is confirmed, the influenced vertices are assigned an index according to their sign relative to the cutting plane. When the object has connectivity, e.g. a triangle mesh, the connectivity is used to propagate the indices to the whole segment, otherwise a restricted K-Nearest Neighbor is used, propagating the index to the nearest vertices whose connection lies inside the Visual Hull of the shape. Whenever the two propagation fronts meet (like, for example, when the user cuts only one part of the handle of a mesh with genus more than 0, Figure 3.7), the propagation interrupted and the user is asked to perform another cut on the handle until a segment is extracted with no ambiguities.

3.2.2 Experimental results

Figure 3.8 shows some example segmentation obtained from the described procedure. The user interaction is limited to a maximum of two cuts per segment border, and the automatic view selection limits the amount of rotation needed by the user for a satisfactory result. Differently from other methods,

Figure 3.8: Some example segmentations obtained thanks to the procedure

the approach is intended to be primitive-independent to be suitable for both surface meshes and point clouds, as tools for manual segmentation the latter are quite rare. The need to avoid any kind of surface reference is the main reason why the cut is completely user-defined.

Limitations At the current implementative state, the algorithm just relies on the correct choice by the user. Humans are however imperfect, and while the *meaning* of their choice is surely the best available, the precision may be defective. What happens is that the cuts are well-positioned to the best of the user capabilities, but don't exactly coincide with the shape concavity due to drawing imperfections, defective view choice or resolution artifacts. Future improvements can include an image-based cut adjustment, using the user input as a *hint* on where to find the best plane according to the perceptive principles. Moreover, an accurate study on the user behavior is needed to find out whether there could be a different, more comfortable way of choosing the cuts on an image.

3.3 Estimating the curvature via skeletal cuts

As shown in the previous section, despite the theoretical robustness of the Minima Rule there is at the moment small room for completely automatic perception-driven algorithms, at least in the practical sense. A different trend can however be found in literature: it is not unusual to associate segmentations and skeletons, both in extracting the descriptor from a partitioning or defining a decomposition from the curve-skeleton of the object. The reason why skeletons have a direct correspondence with shape segmentation is once again found in the perception psychology, and exposed most notably in the work of Biederman [6] and his theory of **Recognition By Component** (RBC): it states that a set of generalized-cones components (called *geons*), robustly detected independently on position, rotation and occlusion, is used for a component-wise recognition of the whole object, such that each component,

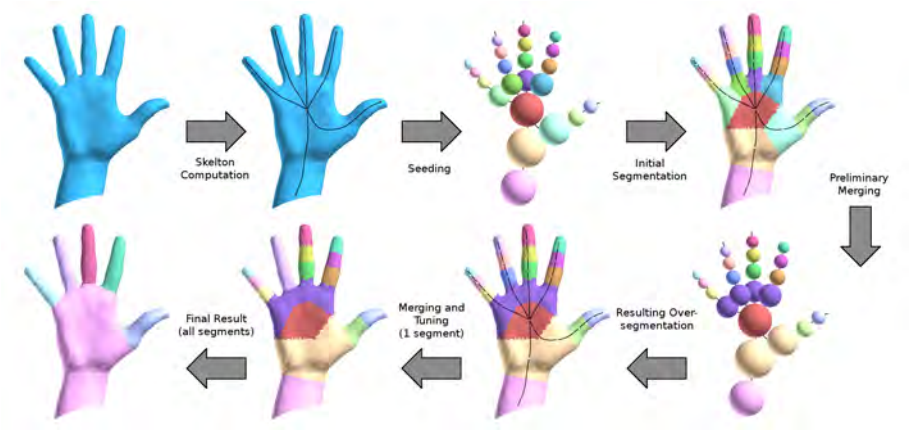


Figure 3.9: An overview of the algorithm: the object's skeleton is computed and seeded. A preliminary merging of the intersecting seeds is performed and the mesh is over-segmented. The merging and tuning algorithm is performed starting from each endpoint, ending when every seed has been processed

or *part*, is directly related to the originating generalized cone. It is then easily understandable that the union of the centers of such cones, that is, the *medial axis* of the shape, directly relates to the segments it is composed of. The direct skeleton-segment relation is then explored in many different ways in literature, be it from inverse distance transform reconstruction [95], plane sweeping [17], geodesics-based [97] or direct vertex-node correspondence [3] among the others. As usual, there is no commonly accepted standard for the definition, and every algorithm shows its pros and cons dependently on the set of objects used, the application and so on.

In order to remark the importance of perception-based algorithms, we propose a method that uses the curve-skeletons described in Chapter 2 for the segmentation. As the cited descriptor is obtained by the direct influence of generalized cones centers, it should reflect the principles of RBC and allow for a robust segmentation suitable for object recognition while maintaining the advantages of the paradigm proposed in this thesis.

3.3.1 Overview

The skeleton-surface relation is a strong tool to overcome the dependence on surface curvature. There is however no generally valid rule for inferring a shape partitioning from its skeleton, and mostly all algorithms define their own approach based on the skeletal structure and the data it provides, leaving the topic still open for research. We propose here a valid alternative for the segmentation problem, mainly for the independence on the data representation thanks to the application of the skeleton extraction method derived from Chapter 2. The different phases of the algorithm, shown in Figure 3.9, will be

discussed in detail in the next Section. What follows is a high-level overview of the whole approach.

The algorithm is based on a region-merging strategy: after the skeleton is computed, it is subsampled into **seeds** according to the radii of the balls; as the skeletal branches meeting into a common joint have no correspondence with a generalized cone axis, the skeletal joints act as starting seeds for the subsampling: proceeding to the endpoints of the skeleton, a node is marked as a seed if and only if its Zone of Influence (ZI) (see definition in Section 2.2.5) is not intersecting the ZI of the previous seed node (see Figure 3.9, third image). Each seed corresponds to a different **supersegment**, and each shape primitive is assigned to the nearest sphere (Figure 3.9, fourth image). Notice however how the ZI intersections are computed only according to the previous node in the graph: nodes belonging to different skeletal branches may belong to the same perceptual *part*, as in the knuckles of the hand, where the four seed spheres are intersecting. A *preliminary merging* step (Figure 3.9, fifth image) is performed for every two seeds with intersecting ZIs. The corresponding segmentation is updated accordingly, and the final **oversegmentation** (Figure 3.9, sixth image) is used as starting point for the **merging** procedure. Starting from each endpoint, the supersegments are *merged* whenever a condition is satisfied (details can be found in the next subsection), or *cut* to create a final segment (Figure 3.9, seventh image) thanks to a *tuning* step that refines the border between the two sections. The algorithm is then reiterated from each endpoint to the center of the mesh until all supersegments have been processed and the final segmentation is computed (Figure 3.9, eight image).

3.3.2 Details and implementation

Merging

Each supersegment is considered to be a subset of the desired segmentation. This means that the final result should be obtained by merging those supersegments that don't correspond to a *border* between two object parts. However, particular care must be put into the merging strategy, as the number of segments isn't necessarily equal to the number of branches, and the joint regions in an object, being the union of several generalized cones, need an ad-hoc approach for a meaningful partitioning.

The only safe assumption is that each endpoint corresponds to a different segment, thanks to the endpoint's *perceptual significance* principle exposed in the skeleton extraction algorithm. Then, starting from an endpoint towards the *core* of the object, each encountered supersegment is processed for merging or cutting.

A distance measure called *Projected Area Distance*(PAD) is computed between the two neighboring segments: let S_i and S_j be the sets of surface elements assigned respectively to the neighboring supersegmnets given by seeds i and j, and $\mathbf{n}$ the vector joining the seed centroids; let A_i and A_j be the areas of the projections of S_i and S_j into a plane along the direction $\mathbf{n}$; the PAD

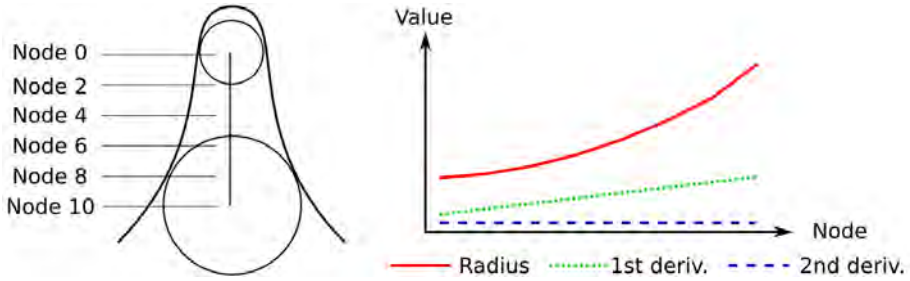


Figure 3.10: A 2D example of a *false positive*: the two skeletal nodes have a strong PAD. However, the shape radius is smoothly increasing, indicating the absence of a cut between the supersegments.

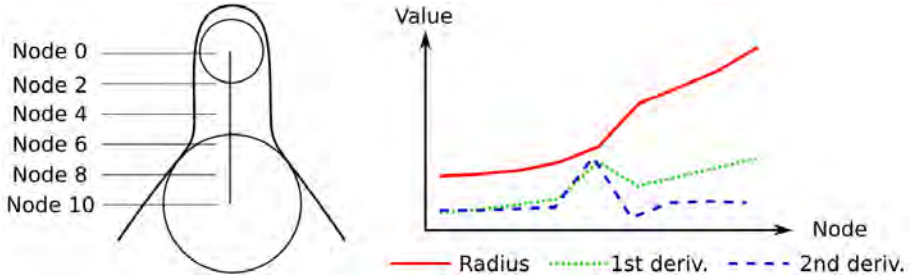


Figure 3.11: A 2D example of a *true positive* with relative cut tuning: the shape radius makes a strong *jump* indicated by a peak in the second derivative. The cut is detected and positioned on the node causing such peak.

PAd_{ij} is then defined as the ratio of the biggest and the smallest value

$$PAd_{ij} = |\max(A_i, A_j)/\min(A_i, A_j)| \tag{3.1}$$

Whenever the measure is below a certain *merging threshold* T_m, the segments are merged and the computation proceeds to the next seed. Otherwise, the distance is a strong indicator that the shape should be *cut* between the two seeds; however, due to the skeletal subsampling, the segmentation is too coarse to be considered meaningful and can give false positives (see Figure 3.10 for a 2D example) that must be avoided. The **cut tuning** procedure described in Subsection 3.3.2 aims at finding the best border between the segments and detecting the false positives. When a segment is cut and confirmed, the next seed is considered as a new endpoint for the reiteration of the merging algorithm.

Joint processing The algorithm is *endpoint-driven*, associating a segment to each endpoint, *cutting* the relative branch whenever the conditions suggest to do so and proceeding towards the only direction possible in the branch. When dealing with joints, due to the different possible directions and their importance in the segmentation, special care is needed. Whenever the merging

procedure reaches a joint segment, the cut-tuning procedure is triggered in any case. The joint is then processed as a starting seed if and only if all its outgoing branches have been processed except for one: in this case, the joint is considered to be a new endpoint and the algorithm goes on as usual until all segments have been visited. When this happens, the segmentation is completed.

Cut-tuning

When the merging threshold T_m triggers the tuning procedure, or a joint has been reached, two situations can arise: a cut is actually present in the shape and the algorithm must find the proper plane to segment the object, or the two supersegments define a **false positive**. Shown for clarity in Figure 3.10, a false positive is given by the fact that the distance formula PAd_{ij} is oblivious to the surface trend and only uses the ratio of the plane projections: this means that a cone-like shape with an elevated surface-skeleton angle would exceed the threshold even when the surface is of constant curvature. The tuning step is then responsible for the detection of such false alarms. Let M be the number of skeletal nodes comprised between the two *seeds* i and j, ordered according to the graph adjacency. Each skeletal node $x = 1 \dots M$ is taken as the defining position for a cutting plane, in a manner similar to the merging step, defined by the point p_x and the normal direction $n_x = p_{x+1} - p_x$; the object-plane intersection area $A(x)$ is computed and stored. In order to recover an approximation of the curvature, we look for the second derivative of such areas with respect to the x (see Figure 3.11); a *jump* in the surface curvature is reflected in a peak of the second derivative, and is thus an indication of the presence of an actual segmentation point. We then define a second threshold, the *tuning threshold* T_t. The tuning distance td_ij is defined as

$$td_t = \frac{\max_{x=1\dots M} A''(x)}{A(\arg\max_{x=1\dots M} A''(x))}$$

that is, the ratio of the maximum second derivative with the value of the area function in the same point. When the measure is below the threshold, the segments are merged and the algorithm returns to the merging step as if the merging threshold hadn't been exceeded at all. Otherwise, a new segment is created at the node $x = \arg\max_x A''(x)$.

In order to keep the whole algorithm compatible with all kinds of representation, particular care has to be put in the cut area computation. Whenever the object provides a surface to work on, the resulting cut is a closed curve whose area is easily computable, but for point clouds and other kinds of representations this is not always the case.

Contour-based area estimation An alternative approach is to estimate the area by looking at the contour of the influenced object parts from different points of view. This techniques is based on the assumption that the object is

Figure 3.12: Some segmentations obtained by the described framework. The results are defective due to meaningless cuts (Dog model), oversegmentation (Octopus model) or undersegmentation (Horse model) according to the parameters, but can be considered as interesting preliminary results for a future improvement of the approach.

describable as a set of *generalized cones* (introduced in Chapter 1), as it is a mandatory precondition for a correct skeleton extraction.

As previously discussed, contours are sufficient for the definition of a segmentation thanks to the Minima Rule and the Shortcut Rule. The overall idea is to *look* at the area between the seeds to find where to cut by analyzing the projected image. In order to reduce any possible ambiguity, only the interested supersegments are rasterized and the *best* view is chosen so that, similarly to the *manual segmentation* framework described in the previous section, the primitives are most scattered onto the image's x axis, and each candidate cutting plane lies horizontal in the image. This results in an efficient way to estimate the cone radius, calculated as the width, in pixels, of the shape in the row given by the cut projection.

3.3.3 Results

Figure 3.12 shows some results obtained by the proposed method. While the algorithm seems to work for simple cases like the hand, the set of parameters makes the whole approach very unstable at the current implementative state. Outside of the threshold parameters, that need a particular study to find the optimal values for a complete set of shapes, other parameters like the resolution of the image on which the cut radii are computed strongly influence the outcome. The Octopus model, for example, shows an oversegmentation on its tentacles due to failures in the tuning computation, due to the excessive narrowness of the tentacles and the consequent errors in the rasterization. Future improvements to the algorithm must focus on the robustness of the whole approach, and a study on how to automatically choose the best values for the thresholds. On a high level, anyway, the algorithm is quite satisfactory and the results, while in a preliminary phase, show that the perceptual approach can obtain interesting outcomes when the conditions are favorable. In terms

of quality of the cuts, the algorithm is strictly dependent on the quality of the skeleton: all the limitations and properties described in Chapter 2 apply to the resulting segmentation. An example is found in the ears of the dog, joined to the head due to the absence of a branch representing them.

3.4 Conclusions and discussion

The act of partitioning is something that has been studied in detail in different fields, as it is an instinctive approach in human life and has important and significative advantages in the computational world. Speaking of shapes, however, partitioning and segmentation become something of higher importance, as the mere *subdivision* process has to include elements of *knowledge*. Perceptual psychology posed strong bases for the field of shape segmentation, so that nowadays principles like the Minima Rule are well known and taken into account by all of the segmentation algorithms. Actually, as shown in this chapter, the perceptual suggestions haven't been directly utilized into the field of Geometry Processing, but served only as an inspiration: contours, the basic elements for shape recognition in humans, play no role in any of the previous works, and the geometric Minima Rule shares only name and goal with its perceptual counterpart.

Stability and significance The way to a stronger significance in the results must take into account human behavior and the perceptual elements. The proposal in Section 3.2 shows how a simple contour-based framework allows a user to specify a segmentation for all kinds of shape representation indiscriminately. Why should partitioning be based on data that are representation-dependent? Similarly to the skeletal extraction method, the perceptual approach poses itself as an alternative to traditional approaches, trying to overcome the limitations of a purely geometric framework.

Implementative limitations and future improvements Unfortunately, at the current stadium there are many limitations to the proposed works. It is however the author's opinion that such limitations are purely implementative. As long as humans are able to segment a shape by its contour, there must be a way of reproducing the innate algorithms in a computational system, and the first time approach given by this chapter shows anyway some potential results with possibility of improvements. Again, the aim of this work is to create an inspiration to future developments, showing the potentiality of the perceptual paradigm.

Chapter 4

Shape reconstruction

> reconstructing (**transitive verb**): to construct again: as **a:** to establish or assemble again **b:** to subject (an organ or part) to surgery to re-form its structure or correct a defect **c:** to build up mentally
>
> The Merriam-Webster Dictionary

In the various fields related to Computer Graphics, the term **reconstruction** refers to the procedure of computing the originating object from a series of measurements. A notable example is the CAT scans, where different *slices* are then reunited into a 3D volume that can be used for further analysis. When speaking of Shape Analysis, *shape reconstruction* aims at extracting the surface of an object from a point cloud and dealing with the issues coming from missing data, noise and so on.

The importance of a correct reconstruction is manifold: since many shape processing algorithms expect a noise-free and watertight mesh, constructing a surface with no holes or artifacts can avoid the need for intermediate cleaning steps as hole-filling, while for surface-based analysis the desired output should faithfully represent the underlying structure of the point cloud in order to derive correct measurements for shape understanding. This is not a simple task: small details and features need to be detected and handled, while still guaranteeing an overall smoothness, and if we take into account also practical challenges as memory and time complexity, one can understand why the reconstruction problem is so important and so differently approached in literature, as pointed out by the heterogeneity of the works cited in the next section.

In Section 4.2 are introduced the main challenges that make shape reconstruction an open research topic, and an introduction on the effects of the perceptual paradigm can have on the problem. We then propose a reconstruction model in Section 4.3 and a practical algorithm for its computation in Section 4.4. The final section shows the results obtained by the proposed approach and a summary of its strengths and weaknesses.

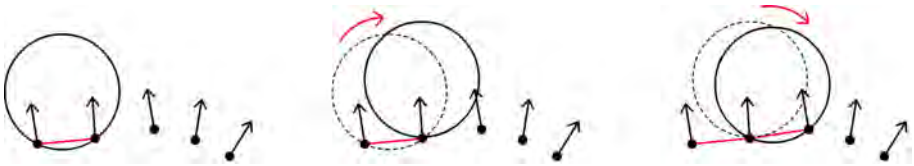


Figure 4.1: The Ball Pivoting Algorithm in 2D. A ball is placed on the points (left) and rotated (center) until it reaches another point (right)

4.1 Previous works

There are many approaches to the reconstruction problem: some try to define the underlying surface in an analytical way, as an isosurface of a 3D function, or directly approaching the points for a procedural reconstruction. Every alternative has its advantages and disadvantages, and of course there is no definable *best* method, as the output quality may differ from dataset to dataset according to some characteristics that are better or worse handled by each individual method.

Ball Pivoting Algorithm In 1999 Bernardini et al. [4] proposed an interesting procedural method for the reconstruction of dense point clouds called **Ball Pivoting Algorithm** (BPA). The BPA is rather intuitive and simple; if the normal for each point is known, a large enough sphere is put *on top* of the shape simulating its collision behavior. The sphere stops on three vertices (see Figure 4.1(left) for the 2D example), forming a triangle, and is then *pivoted* around one of the three border edges (Figure 4.1(center)) until it *hits* another point, thus creating a new triangle adjacent to the previous one (Figure 4.1(right)). By iterating the procedure for each border edge, for a point cloud dense enough the shape is easily reconstructed. Normals are required for the initial placing process and to avoid spurious computations in case the sphere *falls through* the points due to coarse sampling.

Radial Basis Function While the algorithmic approach of the Ball Pivoting returns nice results for well-sampled, clean shapes, Carr et al. [14] proposed to approach the reconstruction problem by fitting an implicit function to efficiently overcome the issues coming from noise, missing data or non-uniform sampling. By considering the shape as the zero-isosurface of a signed distance function, the authors try to interpolate the desired function using a set of **Radial Basis Functions** (RBF), that is, a linear combination of *basis functions* $\Phi_i : \mathbb{R}^3 \times \mathbb{R}^3 \rightarrow \mathbb{R}$:

$$f(x) = \sum_{i=1}^{m} \alpha_i \Phi_i(x, c_i)$$

where $c_{ii=1\ldots m}$ denotes a set of *centers*, and $\alpha_{ii=1\ldots m}$ is the unknown set of *weights*. While already explored by other authors [94] [110] as a mean of surface repre-

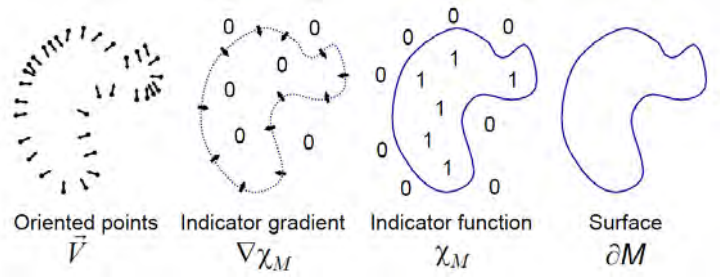


Figure 4.2: An intuitive illustration of the Poisson reconstruction framework, taken from the original paper [48]

sentation, the applicability to real-world data reconstruction was never considered due to the high spatial and time complexity: the high number of *centers* in which to compute the distance function, one for each vertex and a set of *off-surface* constraint to avoid the trivial zero solution, makes the computation unreasonable. Carr et al. however introduce a *center* reduction step for data compression within a desired *fitting accuracy*. The approach has been then reproposed by Samozino et al. [92] using the vertices to construct a Voronoi diagram whose *poles* are used as centers for the RBF interpolation, in order to reduce the needed number of centers due to the stability of the Voronoi poles. In either cases, the off-surface constraints need to be given a *sign* according to the point normals.

Poisson Surface Reconstruction Kazhdan et al. [48] proposed a different approach to shape reconstruction called Poisson Surface Reconstruction (PSR). Given an *indicator function* χ for a shape, having a constant value of 1 for internal points and 0 for external points, has a gradient vector field that's zero almost everywhere, except at the surface. Thus, it has a strong relationship to the surface normals of the shape. The idea is then to find the function χ whose gradient best approximates the surface normals (Figure 4.2). However, as the indicator function is piecewise constant and the vector field would then have unbounded elements at the surface boundary, it is smoothed by a Gaussian filter whose variance is of the order of the sampling resolution. The smoothed vector field $\vec{V}$ represents an approximation of the gradient $\nabla\chi$; however the estimated indicator function $\tilde{\chi}$ cannot be directly computed from the equation $\nabla\tilde{\chi} = \vec{V}$, as the vector field is generally not integrable and there is no unique solution. By applying the divergence operator, a Poisson equation $\Delta\tilde{\chi} = \nabla\cdot\vec{V}$ is formed and a least-squares approximation is applied to find the best solution.

4.2 Challenges in reconstruction

In a perfect world, a set of points describes the object surface with no ambiguities, no missing data and no noise. Unfortunately this is seldom the case: the different methods of obtaining a point-set may return noisy data due to poor sensor resolution; large portions of the shape may be unacquirable due to occlusions or strong concavity; range scans may be poorly aligned due to numerical instability, and so on. Having to deal with such imperfections is of vital importance and most of the previously cited method do exhibit robustness in that sense to a certain extent, both in terms of direct robustness (like the PSR, where missing data is inferred smoothly from the surrounding points) or with ad-hoc steps.

However the main challenge remains the definition of what's *inside* and what's *outside* of the object. The reconstruction algorithms presented in the previous Section, in fact, are all based on point normals to precisely determine what portion of the space is outside of the described shape; unoriented point sets form an almost completely different field. However, the majority of the approach aim just at defining an inside/outside function, or estimate the point normals, in order to apply other known algorithms for shape reconstruction. This is where the perceptual approach does its part.

Perceptual approach Recall the point clouds shown in Chapter 2, Figure 2.18. It is immediate that the represented object on the left is a hand, even if what we actually see is just a set of displaced points, with no real indication of the boundary between object and background. In fact, the human eye perceives surfaces and connectivity where the elements have a strong spatial proximity [33]. Intuitively, it should be enough to *look* at a shape from different points of view to determine what portions of space are part of the object and what are not: the same paradigm applied in Chapter 2 offers an innovative and simple approach to the solution of a shape analysis problem. Our proposal, discussed in Section 4.3, is based on the definition of **Depth Hull**, an extension of the *Visual Hull* introduced in Chapter 2.

4.3 The depth hull in shape understanding

The human visual system is extremely fast and accurate in understanding the spatial relations of a dense enough point cloud. Spatial proximity is seen as a strong indication of an underlying surface. Many image-based reconstruction algorithms like Space Carving [51] are based on the same principle: the object is detected in an area of interest, and what falls outside such area is marked as *outside* also in the resulting reconstruction. This is a direct application of the Visual Hull definition.

An extension of such definition, called the Depth Hull[1], has been derived as a mean of reconstructing an object starting from a series of *depth images* [10].

[1] A detailed discussion on the Depth Hull is found on Appendix A

In the original paper, it's demonstrated that it is the **best approximation** of the object obtainable from the depth images, and for a small number of cameras it allows for a real-time reconstruction. In practice, the number of *depth cameras*, or Z-Cams, that can be placed around the object is limited due both to their cost and the physical space occupation.

4.3.1 Indicator function estimation

It is then possible to reconstruct the Depth Hull of the point cloud by simulating a set of Z-Cams by checking the *depth buffer* of each rendered scene. The possibility of placing a virtually infinite number of Z-Cams around the object allows for a fast and accurate estimation of the *inside* of an object. We then define an indicator function for each voxel according to the *depth values* of each Z-image, given N Z-Cams $C_1 \dots C_N$ and the corresponding functions $D_i(x, y)$ returning the depth value of the i-th Z-image in the pixels x,y. Let $\mathbf{M_i}$ be the transformation matrix from the object coordinates to the framework of the i-th Z-Cam, and, given a point $\mathbf{p}$ in object-space, let $\mathbf{p}^i = \mathbf{M_i}\mathbf{p}$ be the transformed point in the i-th Z-Cam space. The function is defined as

$$X(\mathbf{p}) = \begin{cases} 1, & \text{if } \mathbf{p}_z^i > D_i(\mathbf{p}_x^i, \mathbf{p}_y^i)\ \forall i \in [1 \dots N] \\ 0, & \text{elsewhere} \end{cases}$$

where the subscript is just a coordinate selection $\mathbf{p}_x = \mathbf{e}_x \cdot \mathbf{p}$. In simpler terms, each point is projected on an image and if its Z-value is higher than the depth of the corresponding pixel, then the point is behind the *umbra* of such image. If the condition is satisfied for each camera, then it meets the Depth Hull definition, meaning that the point has to be considered *inside* the object. For a dense point cloud, the described indicator function accurately reflects the Depth Hull of the shape and, consequently, a very good approximation of the underlying object.

4.4 Depth Carving Algorithm

A direct sign and confidence computation for each voxel in a regular grid may be very expensive for high grid resolutions. The usage of an adaptive Octree is the best solution, but some time can be saved by aiming to process only *external* cells in the grid.

The **Depth Carving** algorithm we propose takes inspiration from the Space Carving [51] algorithm used in image-based reconstruction: the sign computation is restricted to the external, *visible* voxels thanks to a *plane sweep* approach. The theoretical complexity is lowered to $O(E \times k)$ where E denotes the *external* voxels in the shape.

Let G be a regular grid composed of $Nx \times Ny \times Nz$ voxels initialized with value 1, enclosed by the bounding box of the point cloud, and let $G_{x,y,z}$ be the voxel cell at coordinates x, y, z. Let B be the set of *border* voxel, i.e. those

voxels with at least one empty neighbor. Starting from an arbitrary point of view, each border voxels is *tested* by reprojecting its centroid into the Z-Image in that direction: if the voxel is *in front* of the Z-Image (i.e., its reprojected depth value is less than the one stored in the image) its value is set to 0, the cell is removed and its 6−neighbors, if not already included in B, are added to the set. When every border voxel has been tested, the process is reiterated from a different point of view until each of all the viewpoints have been considered.

4.4.1 Implementative solutions

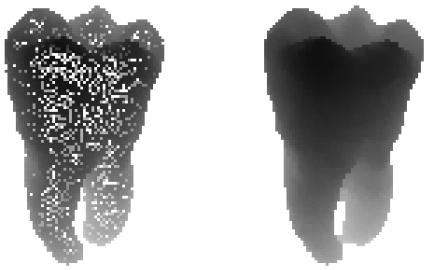


Figure 4.3: Missing depth data (left) can be recovered by a morphological opening of the depth image (right)

As previously said, the Z-Cams can be simulated by synthetic cameras in a 3D visualization framework, using the GPU **depth buffer** to obtain the Z-Images. However, some care must be put into the implementation to overcome issues resulting from this approach. The point clouds may not be dense enough to obtain a smooth depth image. This means that the empty space between the vertices will be *seen* by the camera and carved, resulting in an incorrect reconstruction. This issue can be addressed by lowering the projection resolution or performing a **morphological opening** on the Z-image: this step efficiently simulates a view-dependent Z-splatting (shown in Figure 4.3, right) by smoothing the peaks in the Z depth function, so that small *holes* in the depth image are covered by the values of the neighboring vertices, while maintaining an accurate contour of the shape.

4.5 Results

In this section we show some result obtained by our metric. The meshes shown in Figure 4.4 are isosurfaces extracted from the sign estimation with a simple Marching Cube algorithm.

For dense clouds with no missing data, the reconstruction is accurate enough. Even if the resulting shapes aren't visually pleasing, it must be

Figure 4.4: The results show that dense point clouds are correctly interpreted

considered that an isosurface taken from a boolean space cannot guarantee smooth features. However, the method is intended as an aid for other algorithms as the Radial Basis Function or the Poisson, where a *signing* step is required. Our claim is that the proposed approach is suitable as a lightweight signing tool for unoriented point sets, thus extending the applicability of the cited methods without the need for normal estimation.

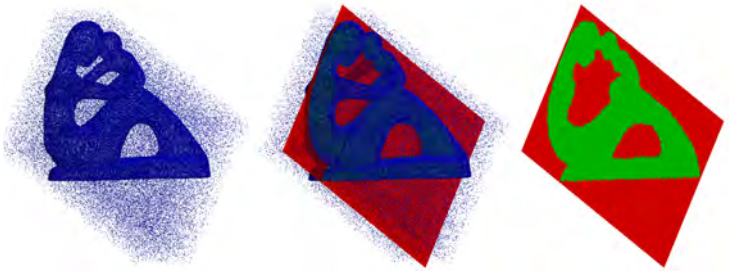


Figure 4.5: The algorithm is robust to white noise, as shown by the cross-section of the indicator function (right image). Internal cells are marked in green, external cells are marked in red

The approach is very robust to noise, as shown by Figure 4.5. Thanks to the multi-view approach, randomly scattered points cannot cast a consistent set of *umbrae*, and the space around them is then carved. The process may detect structured, dense outliers forming a solid shape, as the algorithm has no information on the target shape: everything that may resemble a solid object is reconstructed.

Timing The main advantage of this approach is the independence on the primitive numbers, as the vertices are a factor only in the time needed for rasterization. The whole approach is then fast and efficient. We compare to the signing step in Muller et al. [79], a primitive-based algorithm, showing in Table 4.1 that our method is faster and more robust to changes in the sampling fineness.

Object	Vertices	Muller et al.	Depth Hull
Mug	798,877	42.878	15.126
Fertility	241,607	195.631	4.956
Raptor	49,995	311.416	1.065
Tooth	9,337	3.348	0.467

Table 4.1: Comparisons between our method and the *sign guessing* step in Muller et al. [79] with an adaptive Octree of level 8

4.5.1 Limitations

Figure 4.6: Reconstruction from a *holed* model results in an excessive carving inside the shape. On the left, the depth image of the Bunny model shows missing data in its base (lighter colors denote farther points); on the right, the isosurface shows a set of cavities given by the depth carving

The main limitation comes from the presence of missing data. Wherever the cloud is not dense enough, the carving reaches the other side of the shape leaving an erroneous sign estimation (Figure 4.6); the algorithm, in its present state, cannot detect this absence and interprets the holes as strong concavities. Future improvements to this approach should take into account this issue in order to be applicable to all kinds of datasets.

4.6 Conclusion

A whole lot of complex computations are involved nowadays in understanding a point cloud. As all other cases in the field of Shape Analysis, the overabundance of data drives the algorithm designers towards paths that are oblivious of our real capabilities. The problem of shape understanding is, in the author's opinion, definitely simpler than it looks, and the perceptual elements introduced in this Chapter suggest that more effort should be put in thinking about

how we, as humans, approach a problem before trying to infer results from the data. The method is flawed and limited, but its simplicity suggests that this is not a problem of the approach, but a problem of this particular implementation. The complete novelty, and consequently the absence of previous works to take inspiration from, leave the approach open to flaws that only the experience can solve. Future improvements can and should be done, but the author feels satisfied with the preliminary results as they demonstrate that the perceptual approach can have a significant impact in future developments of this field. The algorithm proposed in this chapter has been accepted for publication in [34].

Chapter 5

Conclusions and discussion

In a field where computing power and over-abundance of data are the main protagonists, the focus to human instinctive processes has been lost in favor of a heavy mathematical computation.

Of course it's not so simple. The proposed algorithms and the suggestions for future work offer an interesting point of view, but the approach is far from being perfect. It is, however, a good starting point for future improvement; a well-thought integration of perceptual elements and data awareness can surely constitute a winning strategy towards a better machine learning.

Effectiveness of the PSA Marr and Nishihara's discussion on shape representation for means of recognition [69] offers a set of criteria that a shape recognition system should comply to. Keeping in mind that Marr's work was mainly focused on human perception and its application in machine recognition, it is easily seen how the PSA approach satisfies the following principles: **accessibility**, **scope and uniqueness**, **stability and sensitivity**. **Accessibility** refers to the possibility of constructing a descriptor starting from an image, and the possibility of doing it inexpensively. While the whole paradigm is image-based, thus satisfying the requisite, the results showed in each chapter demonstrate that the approaches are in most cases faster than, and in the worst cases comparable to, the current geometry-based approaches. Moreover, a geometry-based approach may not be directly applied to real-world cases where the recognizing machine bases its perceptions on one or more cameras. This aspect is discussed in more detail later. **Scope and uniqueness** is the capability of uniquely describing a particular shape, and the focus on a class of objects the descriptor is intended to work on. The best example of this can be found the curve skeleton of Chapter 2: the *scope* of such descriptor is the set of character-like shapes with protruding axes of symmetry, whose Visual Hull is well-defined. According to Marr and Nishihara's theory, the fact that the proposed algorithm won't work on concave shapes like the mug model, as shown in Section 2.3, Figure 2.20, is a *non-limitation*. There cannot exist a universal descriptor for all the classes of shapes, especially when reproducing the

human vision model. **Stability and sensitivity**, already presented in the introduction to Chapter 2, refer to the ability of returning similar descriptors for similar shapes, but still capable of expressing small differences. This criterion is a higher-level characteristic of the descriptors per se, and is satisfied in the instances of segmentation and skeleton extraction. What's more interesting, from a more theoretical point of view, is the discussion on the proposed *design criteria* from Marr and Nishihara's work, showing that the PSA paradigm is extremely suitable to the recognition problem proposed by the two scientists. The **coordinate system** may be *view-centered* or *object-centered*; the authors suggest that a simple recognition task (in their example, a squirrel needing to distinguish trees from other objects) could perform well with view-centered representations, so that even if a different view causes a different descriptor, it would be enough to tell trees and ground apart just by looking at the vertical orientation. Recalling the *shape memory* concept, however, it is immediate to notice how a view-centered system would be unsuitable for complex recognition contexts: a different representation for each vantage point would cause an excess in memory usage, thus suggesting the object-centered model as the better alternative. The perceptual paradigm, while using a multi-view approach, is focused on the definition of an object-centered description. The second design criterion, **primitives**, is of particular interest when we consider that the proposed approach is completely independent on how the shape is represented. In fact, as pointed out by Marr and Nishihara, surface or volumetric primitives may carry information that would be lost by the *early processes* of human vision. The PSA focuses instead on the definition of descriptors that are representation independent, and, most importantly, follow the principles of *early vision* that have been introduced in Chapter 1. The third criterion, **organization**, is of lesser importance, as it seems to be focused on how information is organized in a representation. There is no *correct* way of defining an organization as long as the underlying information is easily available. Well-known descriptors like segmentation and curve-skeletons reflect however what Marr calls *principle of explicit naming*, where the subdivision into smaller and significative elements is vital to the recognition process.

A reliable shape-memory By keeping all computations independent from the object representation, using only the information available to the eye, the PSA algorithms are suitable for a *real world* implementation for automatic vision systems. Even if the algorithms haven't been tested on actual image-based acquisitions, the approach can be used as a mean of constructing a **shape memory** for the system: recalling the principle exposed in Section 1.4, shape recognition is based on a set of pre-stored descriptors that human vision refers to. The possibility of computing descriptors utilizing synthetic, well-defined shapes, may result in a fast and robust way of constructing such memory, while maintaining a compatibility with the cognitive system: the machine, *seeing* a new object, would only be able to compute image-based descriptors, thus increasing the efficiency of the recognition process when the whole shape memory is built using an analogous approach.

Figure 5.1: Visual cues as textures, shadows and depth are secondary with respect to the contours in a recognition context. The figure is easily recognized as an octopus even if the texturing is unnatural

Visual cues The proposed work may seem limited in the definition: if the aim is to take inspiration from the living, and humans have a complex vision system, why the focus is only on contours and monochromatic images? Except for the reconstruction in Chapter 4, where the depth information can be considered an analogy of the stereoscopic vision, the image used in the algorithms lack color or shading which are an important factor in our everyday experience. To explain the choice it's necessary to recall the psychological studies this thesis is based on. The importance of the *visual cues* is acknowledged, but the majority of the works are based on line contours, as demonstrated by Marr to be the key point in human shape understanding. Moreover, by designing a system as simple as possible, allowing it to function nicely with the least amount of data, can only be an advantage in terms of robustness, and the additional information coming from the visual cues can be used to improve the speed of the algorithm or the accuracy of the output, but should not be considered vital for the whole recognition process. Moreover, it reflects once again our behavior. Look at Figure 5.1: despite the unnatural texturing, the shape is recognized as an octopus. Why should then all other visual cues be considered equally to contours in the recognition process?

Concluding remarks The PSA is, in the author's opinion, a promising field. The current implementations of its algorithms are faulty, but the limitations they show aren't relative to the approach itself. The strong link with perceptual psychology suggests that, whenever a *classic* Shape Analysis algorithm tries to extract *knowledge* from a shape, a PSA implementation could be performed. Computations like e.g. the Shape Diameter Function [97] may find a simpler definition in a PSA context, as the thickness of an object is something the human eye can detect easily; motion segmentation, aiming at subdividing a shape by comparing two different poses using surface curvature changes [43], can be approached by restricting the comparison to just those elements that change in

the contour [70], or by comparing the Optical Flow of two subsequent *frames* in the interpolation in a manner inspired from real-world motion detection. Whenever the task to be performed is strictly correlated to some instinctive human capability, the PSA can achieve better results as it tries to imitate those behaviors and *algorithms* that are innate in humans and that pose the basis for the construction of our knowledge.

Appendix A

Computer vision background

Our approach borrows some tools and definitions from the field of *computer vision*, where the bonds with human perception are stronger. As some notions can be unfamiliar to a shape processing audience, we briefly summarize the definitions in this section in order to make the presentation of our approach more self-consistent.

A.1 The Visual Hull

The Visual Hull (VH) was introduced by Laurentini [52] to approach the problem of *shape from silhouettes*, describing how a set of object projections onto an image plane influence its perception, and the assumptions that can be safely made on the object just basing on the information coming from its projection. Let $C \subseteq \mathbb{R}^3$ be a set of points of view; the *visual hull* of an object O relative to C, $\mathcal{VH}(O, C)$, is defined as the subspace of $\mathbb{R}^3$ such that, for each point $p \in \mathcal{VH}(O, C)$, and each point $c \in C$, the projective ray starting at c and passing through p contains at least a point of O:

$$\mathcal{VH}(O, C) = \{\, p \mid \overline{cp} \cap O \neq \emptyset,\ \forall c \in C \,\}.$$

It is immediate to see how $O \subseteq \mathcal{VH}(O, C)$ for every C as each point in the surface satisfies the definition. The Visual Hull is the *maximal object* that is *silhouette-equivalent* to O with respect to C, that is, the maximal object returning the same silhouette as S for each vantage point $V \in C$. In fact,

1. $\mathcal{VH}(O, C)$ is *silhouette-equivalent* to S as the projection of any point of the Visual Hull from any point of view $V \in C$ belongs by definition to the silhouette obtained from V, and the projection of any surface point from any point of view $V \in C$ belongs to the silhouette of the Visual Hull obtained from V since $O \subseteq \mathcal{VH}(O, C)$.

2. $\mathcal{VH}(O, C)$ is *maximal* since for any point $P \notin \mathcal{VH}(O, C)$ there is at least a straight line starting at $V \in C$, passing through P, not intersecting O.

Consequently, the projection of P does not belong to the silhouette of O computed from V, implying that each point $P \notin \mathcal{VH}(O, C)$ cannot be part of an object silhouette-equivalent to O.

In other words, the visual hull of an object is the maximal object that gives the same silhouettes of the original one from any considered viewpoint. $\mathcal{VH}(O, C)$ is also the **closest approximation** of O that we can obtain using its silhouettes, that is, the best representation of what a human viewer would appreciate of the object in absence of other information.

A.2 The Depth Hull

The *Depth Hull* (DH) is a generalization of the VH first defined by Bogomjakov et al. [10] to approach the problem of reconstructing and rendering an image-based geometry from a set of depth cameras, also called Z-Cams. Such cameras are capable of returning depth information for each pixel in the view, and the *umbra* of a viewed scene is the portion of the visual cone behind the depth map, or the shadow cone generated by the object if a point light source is placed in the camera position. The DH of an object O with respect to a set of viewpoints $C \subseteq \mathbb{R}^3$, $\mathcal{DH}(O, C)$ is defined as the intersection of the *umbrae* of the given reference Z-Cams In general, for an object O and a set of viewpoints $C \subseteq \mathbb{R}^3$ located outside of the object's Convex Hull, the following relations hold:

$$S \subseteq \mathcal{DH}(O, C) \subseteq \mathcal{VH}(O, C) \subseteq \mathcal{CH}(O).$$

The Depth Hull is shown to be the best approximation of the geometry of an object, when viewed from a reference set of viewpoints, obtainable by Z-Cams.

A.3 Epipolar Geometry and rectification

Epipolar geometry provides us a quite useful constraint while trying to couple planar images of a point lying in the 3D space in order to find its spatial coordinates (see Figure A.1). Let's suppose to have a point $P \in \mathbb{R}^3$ and two projective cameras $\mathsf{C}_0, \mathsf{C}_1$ (with $\mathsf{C}_0 \neq \mathsf{C}_1$) which respectively project into the planes I_0, I_1. Given the image point $p_0 = \mathsf{I}_0(P)$, the point projected by the secondary camera, $p_1 = \mathsf{I}_1(P)$, must lie in the line defined by the intersection between the plane I_1 and the *epipolar plane* π_P. Namely, the plane defined by the point P and the centers of projection of C_1 and C_2, meaning that for each p_1 the corresponding point p_2 must be searched in a mono-dimensional space instead of a bi-dimensional one.

The epipolar constraint can be enforced with *image rectification* in order to further reduce the complexity of this search: when the epipolar planes are parallel, *epipolar lines* become horizontal and the search is restricted to a *scanline matching*, meaning that is possible to search for a correspondence between the pixels along the same scanline in the two images. While image

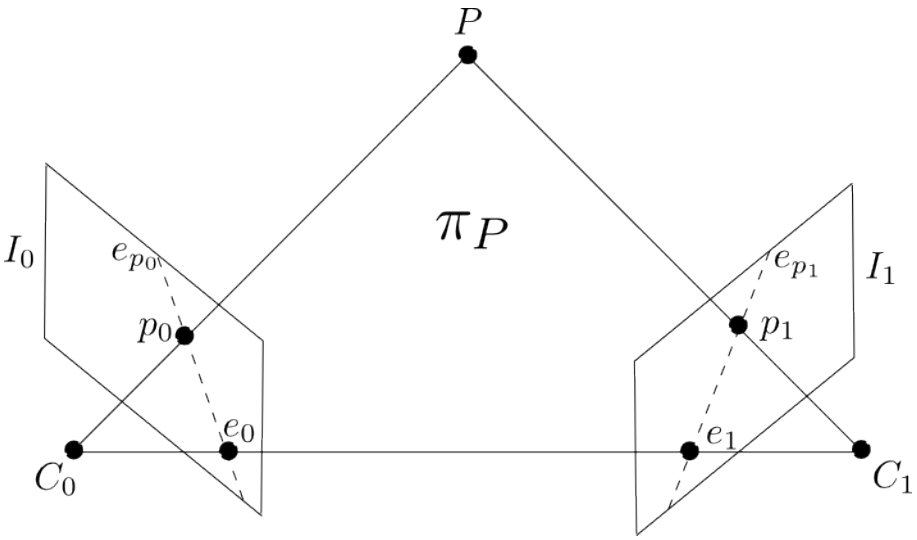


Figure A.1: An example of Epipolar Geometry: cameras C_0 and C_1 see a point P respectively in its projections p_0 and p_1 on the image planes. The intersection between the epipolar plane π_P and the image planes return the epipolar lines e_{p_0} and e_{p_1}, that constitute the epipolar constraint

transformations are employed for rectification in real world cases with projective cameras, this effect is also obtained using **affine projections**: it can be approximated by using a very long focal length camera. In synthetic environment, like the one we present in subsection 2.2.3, the possibility to directly employ parallel projections can increase the efficiency of the matching.

Bibliography

[1] Arbeláez, P. A., and Cohen, L. D. A metric approach to vector-valued image segmentation. *International Journal of Computer Vision 69*, 1 (Aug. 2006), 119–126.

[2] Arcelli, C., Sanniti di Baja, G., and Serino, L. Distance-driven skeletonization in voxel images. *IEEE Trans. Pattern Anal. Mach. Intell. 33* (April 2011), 709–720.

[3] Au, O. K.-C., Tai, C.-L., Chu, H.-K., Cohen-Or, D., and Lee, T.-Y. Skeleton extraction by mesh contraction. In *SIGGRAPH '08* (Aug. 2008), pp. 1–10.

[4] Bernardini, F., Mittleman, J., Rushmeier, H., Silva, C., and Taubin, G. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics 5* (October 1999), 349–359.

[5] Biasotti, S., Marini, S., Mortara, M., and Patané, G. An overview on properties and efficacy of topological skeletons in shape modelling. In *Proceedings of the Shape Modeling International 2003* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 245–.

[6] Biederman, I. Recognition-by-components: A theory of human image understanding. *Psychological Review 94* (1987), 115–147.

[7] Binford, T. O. Visual perception by computer. In *Proceedings of the IEEE Conference on Systems and Control (Miami, FL)* (1971).

[8] Blum, H. A transformation for extracting new descriptions of shape. In *Models for the Perception of Speech and Visual Form* (1967), pp. 362–380.

[9] Blum, H. Biological shape and visual science (part i). *Journal of Theoretical Biology 38*, 2 (1973), 205 – 287.

[10] Bogomjakov, A., Gotsmann, C., and Magnor, M. Free-viewpoint video from depth cameras. In *Proc. Vision, Modeling and Visualization (VMV) 2006* (Nov. 2006), pp. 89–96.

[11] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S. Analysis of emotion recognition using facial expressions, speech and multimodal information. In

Proceedings of the 6th international conference on Multimodal interfaces (New York, NY, USA, 2004), ICMI '04, ACM, pp. 205–211.

[12] Canny, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-8*, 6 (nov. 1986), 679 –698.

[13] Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., and Su, Z. Point cloud skeletons via Laplacian based contraction. In *SMI 2010* (June 2010), pp. 187–197.

[14] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., and Evans, T. R. Reconstruction and representation of 3d objects with radial basis functions. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 67–76.

[15] Chen, J.-H., and Chen, C.-S. Using inter-feature-line consistencies for sequence-based object recognition. In *8th European Conference on Computer Vision (ECCV 2004)* (May 2004), pp. 108–120.

[16] Chen, X., Golovinskiy, A., and Funkhouser, T. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH) 28*, 3 (Aug. 2009).

[17] Cheng, Z.-Q., Li, B., Dang, G., and Jin, S.-Y. Meaningful mesh segmentation guided by the 3d short-cut rule. In *Proceedings of the 5th international conference on Advances in geometric modeling and processing* (Berlin, Heidelberg, 2008), GMP'08, Springer-Verlag, pp. 244–257.

[18] Cornea, N., Silver, D., Yuan, X., and Balasubramanian, R. Computing hierarchical curve-skeletons of 3D objects. *The Visual Computer 21*, 11 (October 2005), 945–955.

[19] Cornea, N. D., Silver, D., and Min, P. Curve-skeleton applications. In *IEEE Visualization* (Oct. 2005), pp. 95–102.

[20] Corney, J. R., Rea, H., Clark, D., Pritchard, J., Breaks, M., and MacLeod, R. Course filters for shape matching. *IEEE Computer Graphics and Applications 22*, 3 (2002), 65–74.

[21] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE 18*, 1 (jan 2001), 32 –80.

[22] Cyr, C. M., and Kimia, B. B. A similarity-based aspect-graph approach to 3d object recognition. *Int. J. Comput. Vision 57* (April 2004), 5–22.

[23] Davidov, D., Tsur, O., and Rappoport, A. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning* (Stroudsburg, PA, USA, 2010), CoNLL '10, Association for Computational Linguistics, pp. 107–116.

[24] Dey, T. K., and Sun, J. Defining and computing curve-skeletons with medial geodesic function. In *Symposium on Geometry Processing* (2006), pp. 143–152.

[25] Eilemann, S., Makhinya, M., and Pajarola, R. Equalizer: A scalable parallel rendering framework. *IEEE Transactions on Visualization and Computer Graphics 15*, 3 (May/June 2009), 436–452.

[26] Elfenbein, H. A., and Ambady, N. On the universality and cultural specificity of emotion recognition : A meta-analysis. *Psychological Bulletin 128*, 2 (2002), 203–235.

[27] Fan, L., Lic, L., and Liu, K. Paint mesh cutting. *Computer Graphics Forum 30*, 2 (2011), 603–612.

[28] Floriani, L., and Spagnuolo, M. *Shape Analysis and Structuring*. Mathematics and Visualization. Springer Berlin Heidelberg, 2008.

[29] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D. A search engine for 3d models. *ACM Trans. Graph. 22* (January 2003), 83–105.

[30] Gagvani, N., and Silver, D. Parameter-controlled volume thinning. *Graphical Models and Image Processing 61*, 3 (1999), 149 – 164.

[31] Ge, F., Wang, S., and Liu, T. Image-segmentation evaluation from the perspective of salient object extraction. In *2006 Conference on Computer Vision and Pattern Recognition (CVPR 2006)* (June 2006), pp. 1146–1153.

[32] Gervais, M. J., Harvey, L. O., and Roberts, J. O. Identification confusions among letters of the alphabet. *Journal of Experimental Psychology: Human Perception and Performance 10*, 5 (1984), 655 – 666.

[33] Gross, M. Are points the better graphics primitives? *Computer Graphics Forum 20*, 3 (2001), xvii–xvii.

[34] Guggeri, F., Scateni, R., and Pajarola, R. Shape reconstruction from raw point clouds using depth carving. In *Eurographics 2012 - Short Papers* (Aire-la-Ville, Switzerland, 2012), Eurographics Association.

[35] Guillemin, V., and Pollack, A. *Differential Topology*. AMS Chelsea Publishing Series. Amer Mathematical Society, 2010.

[36] Hassouna, M. S., and Farag, A. A. Robust centerline extraction framework using level sets. In *CVPR '05* (2005), vol. 1, pp. 458–465.

[37] Hoffman, D., and Richards, W. Parts of recognition. *Cognition 18*, 1-3 (1984), 65 – 96.

[38] Holbrook, M. A comparison of methods for measuring the interletter similarity between capital letters. *Attention, Perception, & Psychophysics 17* (1975), 532–536. 10.3758/BF03203964.

[39] Horn, B. K. P. The binford-horn line finder.

[40] Huffman, D. A. Impossible objects as nonsense sentences. In *Machine Intelligence* (1971).

[41] Ip, C. Y., Lapadat, D., Sieger, L., and Regli, W. C. Using shape distributions to compare solid models. In *Proceedings of the seventh ACM symposium on Solid modeling and applications* (New York, NY, USA, 2002), SMA '02, ACM, pp. 273–280.

[42] Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., and Ramani, K. A reconfigurable 3d engineering shape search system part i: Shape representation.

[43] James, D. L., and Twigg, C. D. Skinning mesh animations. *ACM Transactions on Graphics (SIGGRAPH 2005) 24*, 3 (Aug. 2005).

[44] Ji, Z., Liu, L., Chen, Z., and Wang, G. Easy mesh cutting. *Computer Graphics Forum 25*, 3 (2006), 283–291.

[45] Jolicoeur, P. The time to name disoriented natural objects. *Memory & cognition 13*, 4 (July 1985), 289–303.

[46] Katz, S., Leifman, G., and Tal, A. Mesh segmentation using feature point and core extraction. *The Visual Computer 21* (2005), 649–658. 10.1007/s00371-005-0344-9.

[47] Katz, S., and Tal, A. Hierarchical mesh decomposition using fuzzy clustering and cuts. *ACM Trans. Graph. 22* (July 2003), 954–961.

[48] Kazhdan, M., Bolitho, M., and Hoppe, H. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2006), SGP '06, Eurographics Association, pp. 61–70.

[49] Kazhdan, M., Chazelle, B., Dobkin, D., Funkhouser, T., and Rusinkiewicz, S. A reflective symmetry descriptor for 3d models. *Algorithmica 38* (October 2003), 201–225.

[50] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), SGP '03, Eurographics Association, pp. 156–164.

[51] Kutulakos, K. N., and Seitz, S. M. A theory of shape by space carving. *Int. J. Comput. Vision 38* (July 2000), 199–218.

[52] Laurentini, A. The visual hull concept for silhouette-based image understanding. *IEEE Trans. Pattern Anal. Mach. Intell. 16*, 2 (1994), 150–162.

[53] Lee, C. H., Varshney, A., and Jacobs, D. W. Mesh saliency. In *ACM SIGGRAPH 2005 Papers* (New York, NY, USA, 2005), SIGGRAPH '05, ACM, pp. 659–666.

[54] Lee, Y., Lee, S., Shamir, A., Cohen-Or, D., and Seidel, H.-P. Mesh scissoring with minima rule and part salience. *Comput. Aided Geom. Des. 22* (July 2005), 444–465.

[55] Leek, E. Effects of stimulus orientation on the identification of common polyoriented objects. *Psychonomic Bulletin and Review 5*, 4 (1998), 650–658.

[56] Leung, T. K., and Malik, J. Contour continuity in region based image segmentation. In *5th European Conference on Computer Vision (ECCV 1998)* (June 1998), pp. 544–559.

[57] Lien, J.-M., and Amato, N. M. Approximate convex decomposition. In *Proceedings of the twentieth annual symposium on Computational geometry* (New York, NY, USA, 2004), SCG '04, ACM, pp. 457–458.

[58] Lien, J.-M., and Amato, N. M. Approximate convex decomposition of polygons. *Comput. Geom. Theory Appl. 35* (August 2006), 100–123.

[59] Lien, J.-M., Keyser, J., and Amato, N. M. Simultaneous shape decomposition and skeletonization. In *SPM '06* (2006), pp. 219–228.

[60] Liu, L., Chambers, E. W., Letscher, D., and Ju, T. A simple and robust thinning algorithm on cell complexes. *Comput. Graph. Forum 29*, 7 (2010), 2253–2260.

[61] Liu, R., and Zhang, H. Segmentation of 3d meshes through spectral clustering. In *Proceedings of the Computer Graphics and Applications, 12th Pacific Conference* (Washington, DC, USA, 2004), PG '04, IEEE Computer Society, pp. 298–305.

[62] Livesu, M., Guggeri, F., and Scateni, R. Reconstructing the curve-skeletons of 3d shapes using the visual hull. *IEEE Transactions on Visualization and Computer Graphics*.

[63] Loffler, J. Content-based retrieval of 3d models in distributed web databases by visual shape information. In *Proceedings of the International Conference on Information Visualisation* (Washington, DC, USA, 2000), IEEE Computer Society, pp. 82–.

[64] Lou, K., Jayanti, S., Iyer, N., Kalyanaraman, Y., Ramani, K., and Prabhakar, S. A Reconfigurable, Intelligent 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering. *Proceedings of ASME DETC' 03, 23rd Computers and Information in engineering (CIE) Conference* (Sept. 2003).

[65] Lyu, S. Mercer kernels for object recognition with local features. In *2005 Conference on Computer Vision and Pattern Recognition (CVPR 2005)* (June 2005), pp. 223–229.

[66] Ma, C. M., and Sonka, M. A fully parallel 3D thinning algorithm and its applications. *Comput. Vis. Image Underst. 64*, 3 (1996), 420–433.

[67] Marr, D. Early processing of visual information. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences 275*, 942 (1976), pp. 483–519.

[68] Marr, D. Analysis of occluding contour. *Proceedings of the Royal Society of London. Series B, Biological Sciences 197*, 1129 (1977), pp. 441–475.

[69] Marr, D., and Nishihara, H. K. Representation and recognition of the spatial organization of three-dimensional shapes. *Proceedings of the Royal Society of London. Series B, Biological Sciences 200*, 1140 (1978), pp. 269–294.

[70] Marras, S., Hormann, K., Bronstein, M., Scateni, R., and Scopigno, R. Motion-based segmentation using augmented silhouettes. *Submitted to Geometry Modeling and Processing (GMP 2012)*.

[71] Massaro, D. W. Illusions and issues in bimodal speech perception. In *PROCEEDINGS OF AUDITORY VISUAL SPEECH PERCEPTION '98* (1998), pp. 21–26.

[72] Matsumoto, D. Cultural influences on the perception of emotion. *Journal of CrossCultural Psychology 20*, 1 (1989), 92–105.

[73] McMullen, P. A., and Farah, M. J. Viewer-centered and object-centered representations in the recognition of naturalistic line drawings. *Psychological Science 2*, 4 (1991), pp. 275–277.

[74] Mehalawi, E. M. A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment. *Computer-Aided Design 35*, 1 (Jan. 2003), 95–105.

[75] Mehalawi, M. E., and Allen. A database system of mechanical components based on geometric and topological similarity. Part I: representation. *Computer-Aided Design 35*, 1 (Jan. 2003), 83–94.

[76] Mihalcea, R., and Strapparava, C. Making computers laugh: investigations in automatic humor recognition. In *Proceedings of the conference*

on Human Language Technology and Empirical Methods in Natural Language Processing (Stroudsburg, PA, USA, 2005), HLT '05, Association for Computational Linguistics, pp. 531–538.

[77] Miklos, B., Giesen, J., and Pauly, M. Discrete scale axis representations for 3D geometry. *ACM Trans. Graph. 29* (July 2010), 101:1–101:10.

[78] Moravec, H. When will computer hardware match the human brain.

[79] Mullen, P., De Goes, F., Desbrun, M., Cohen-Steiner, D., and Alliez, P. Signing the unsigned: Robust surface reconstruction from raw pointsets. *Computer Graphics Forum 29*, 5 (2010), 1733–1741.

[80] Murch, R. D., and McGregor, B. P. Reconstituting object shape and orientation from silhouettes. *J. Opt. Soc. Am. A 9*, 9 (Sep 1992), 1491–1497.

[81] Neisser, U. *Cognitive Psychology*. New York: Appleton-Century-Crofts, 1989.

[82] Novotni, M., and Klein, R. 3d zernike descriptors for content based shape retrieval. In *Proceedings of the eighth ACM symposium on Solid modeling and applications* (New York, NY, USA, 2003), SM '03, ACM, pp. 216–225.

[83] Ohbuchi, R., Otagiri, T., Ibato, M., and Takei, T. Shape-similarity search of three-dimensional models using parameterized statistics. In *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2002), PG '02, IEEE Computer Society, pp. 265–.

[84] Ohbuchi, R., and Takei, T. Shape-similarity comparison of 3d models using alpha shapes. In *Proceedings of the 11th Pacific Conference on Computer Graphics and Applications* (Washington, DC, USA, 2003), PG '03, IEEE Computer Society, pp. 293–.

[85] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. Shape distributions. *ACM Trans. Graph. 21* (October 2002), 807–832.

[86] Paquet, E., Rioux, M., Murching, A., Naveen, T., and Tabatabai, A. Description of shape information for 2-d and 3-d objects. *Signal Processing Image Communication 16*, 1-2 (2000), 103–122.

[87] Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G. PSYCHOLOGICAL ASPECTS OF NATURAL LANGUAGE USE: Our Words, Our Selves. *Annual Review of Psychology 54*, 1 (2003), 547.

[88] PENROSE, L. S., and PENROSE, R. Impossible objects: A special type of visual illusion. *British Journal of Psychology 49*, 1 (1958), 31–33.

[89] Petitjean, S. A Computational Geometric Approach to Visual Hulls. *International Journal of Computational Geometry & Applications 8* (1998), 407–436.

[90] Richards, W. A., Koenderink, J. J., and Hoffman, D. D. Inferring three-dimensional shapes from two-dimensional silhouettes. *J. Opt. Soc. Am. A 4*, 7 (Jul 1987), 1168–1175.

[91] Rosenfeld, A., and Thurston, M. Edge and curve detection for visual scene analysis. *Computers, IEEE Transactions on C-20*, 5 (may 1971), 562 – 569.

[92] Samozino, M., Alexa, M., Alliez, P., and Yvinec, M. Reconstruction with voronoi centered radial basis functions. In *Proceedings of the fourth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2006), SGP '06, Eurographics Association, pp. 51–60.

[93] Sanniti di Baja, G. Well-shaped, stable, and reversible skeletons from the (3,4)-distance transform. *Journal of Visual Communication and Image Representation 5* (1994), 107–115.

[94] Savchenko, V., Pasko, E. A., Okunev, O. G., and Kunii, T. L. Function representation of solids reconstructed from scattered surface points and contours. *Computer Graphics Forum 14* (1995), 181–188.

[95] Serino, L., Sanniti di Baja, G., and Arcelli, C. Object decomposition via curvilinear skeleton partition. In *ICPR* (2010), pp. 4081–4084.

[96] Shamir, A. A survey on mesh segmentation techniques. *Computer Graphics Forum 27*, 6 (Sept. 2008), 1539–1556.

[97] Shapira, L., Shamir, A., and Cohen-Or, D. Consistent mesh partitioning and skeletonisation using the shape diameter function. *Vis. Comput. 24*, 4 (2008), 249–259.

[98] Sharf, A., Lewiner, T., Shamir, A., and Kobbelt, L. On-the-fly curve-skeleton computation for 3D shapes. *Computer Graphics Forum 26*, 3 (Oct. 2007), 323–328.

[99] Shneiderman, B. The limits of speech recognition. *Commun. ACM 43* (September 2000), 63–65.

[100] Siddiqi, K., and Pizer, S. *Medial Representations: Mathematics, Algorithms and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2008.

[101] Singh, M., Seyranian, G. D., and Hoffman, D. D. Parsing silhouettes: the short-cut rule. *Perception And Psychophysics 61*, 4 (1999), 636–660.

[102] Spertus, E. Smokey: Automatic recognition of hostile messages. In *In Proc. IAAI* (1997), pp. 1058–1065.

[103] Sundar, H., Silver, D., Gagvani, N., and Dickinson, S. Skeleton based shape matching and retrieval. In *Proceedings of the Shape Modeling International 2003* (Washington, DC, USA, 2003), IEEE Computer Society, pp. 130–.

[104] Tagliasacchi, A., Zhang, H., and Cohen-Or, D. Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph. 28*, 3 (2009), 1–9.

[105] Tangelder, J. W., and Veltkamp, R. C. A survey of content based 3d shape retrieval methods. *Multimedia Tools Appl. 39* (September 2008), 441–471.

[106] Taylor, J. M., and Mazlack, L. J. Computationally recognizing wordplay in jokes. In *In Proceedings of CogSci 2004* (2004).

[107] Tepperman, J., Traum, D., and Narayanan, S. "yeah right": Sarcasm recognition for spoken dialogue systems signal analysis and interpretation laboratory , university of southern california institute for creative technologies , university of southern california. *InterSpeech ICSLP* (2006), 3–6.

[108] Thompson, D., and Bonner, J. *On growth and form*. A Canto Book Series. Cambridge University Press, 1992.

[109] Tsur, O., Davidov, D., and Rappoport, A. Icwsm - a great catchy name: Semi-supervised recognition of sarcastic sentences in product reviews. *ICSWM*, 9 (2010).

[110] Turk, G., and O'Brien, J. F. Shape transformation using variational implicit functions. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1999), SIGGRAPH '99, ACM Press/Addison-Wesley Publishing Co., pp. 335–342.

[111] Ullman, S. An approach to object recognition: Aligning pictorial descriptions. *Laboratory, Massachusetts Institute of Technology 931* (1986), 93–1.

[112] Vasconcelos, M., Vasconcelos, N., and Carneiro, G. Weakly supervised top-down image segmentation. In *2006 Conference on Computer Vision and Pattern Recognition (CVPR 2006)* (June 2006), pp. 1001–1006.

[113] Ververidis, D., and Kotropoulos, C. Emotional speech recognition: Resources, features, and methods. *Speech Communication 48*, 9 (2006), 1162 – 1181.

[114] Vogt, T., André, E., and Wagner, J. Affect and emotion in human-computer interaction. Springer-Verlag, Berlin, Heidelberg, 2008, ch. Automatic Recognition of Emotions from Speech: A Review of the Literature and Recommendations for Practical Realisation, pp. 75–91.

[115] Vranic, D. V., Saupe, D., and Richter, J. Tools for 3d-object retrieval: Karhunen-loeve transform and spherical harmonics. In *IEEE MMSP 2001* (2001), pp. 293–298.

[116] Waltz, D. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision* (1975), McGraw-Hill, p. pages.

[117] Wan, L. Parts-based 2d shape decomposition by convex hull. *2009 IEEE International Conference on Shape Modeling and Applications* (2009), 89–95.

[118] Wang, T., and Basu, A. A note on 'A fully parallel 3D thinning algorithm and its applications'. *Pattern Recogn. Lett. 28*, 4 (2007), 501–506.

[119] Warrington, E. K., and Taylor, A. M. The contribution of the right parietal lobe to object recognition. *Cortex; a journal devoted to the study of the nervous system and behavior 9*, 2 (June 1973), 152–164.

[120] Warrington, E. K., and Taylor, A. M. Two categorical stages of object recognition. *Perception* (1978), 695–705.

[121] Witkin, A. P. Scale-space filtering. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 2* (San Francisco, CA, USA, 1983), Morgan Kaufmann Publishers Inc., pp. 1019–1022.

[122] Yim, P., Choyke, P., and Summers, R. Gray-scale skeletonization of small vessels in magnetic resonance angiography. *Medical Imaging, IEEE Transactions on 19*, 6 (June 2000), 568–576.

[123] Zhang, C., and Chen, T. Efficient feature extraction for 2d/3d objects in mesh representation. *Virtual Reality 3* (2001), 1–4.

[124] Zhang, C., and Chen, T. Indexing and retrieval of 3d models aided by active learning. In *Proceedings of the ninth ACM international conference on Multimedia* (New York, NY, USA, 2001), MULTIMEDIA '01, ACM, pp. 615–616.

[125] Zhang, X., Liu, J., Li, Z., and Jaeger, M. Volume decomposition and hierarchical skeletonization. In *VRCAI '08* (2008), pp. 1–6.

[126] Zuckerberger, E. Polyhedral surface decomposition with applications. *Computers and Graphics 26*, 5 (Oct. 2002), 733–743.